STAT 466 HW 10

Brody Anderson

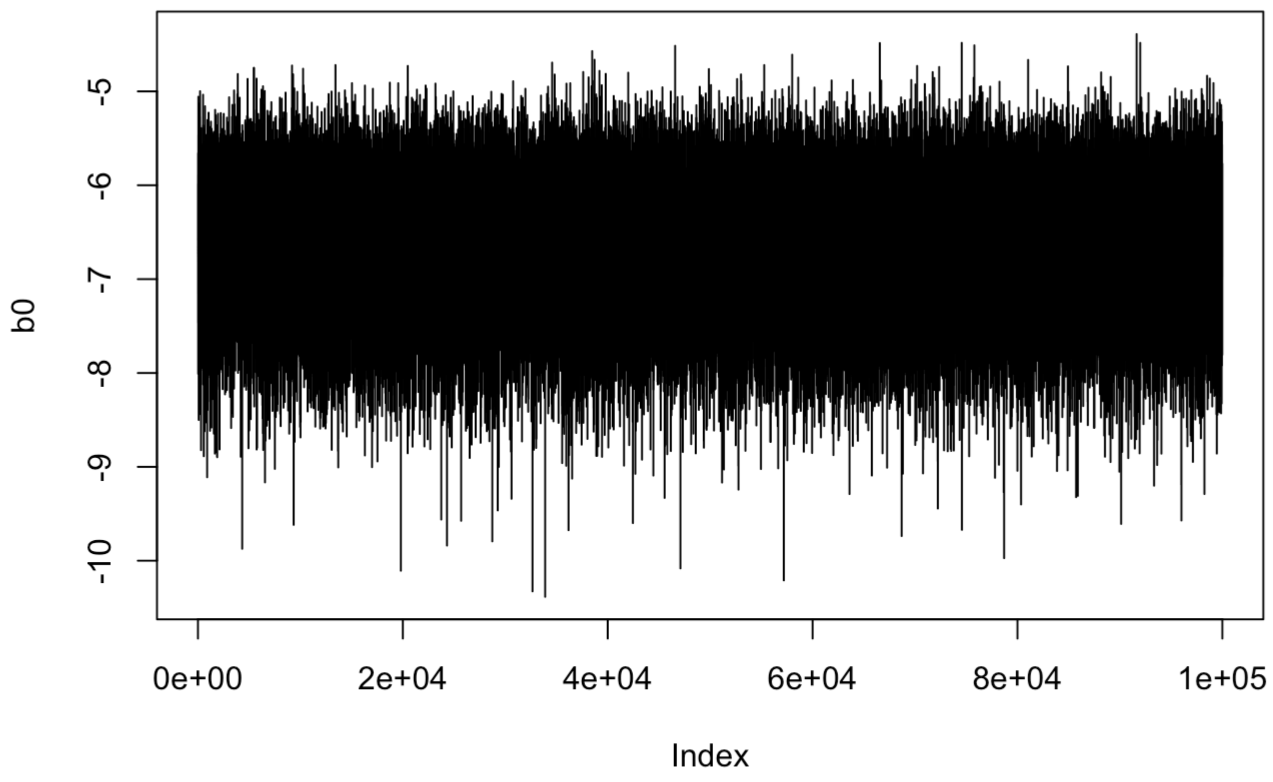
2024-03-22

Question 1

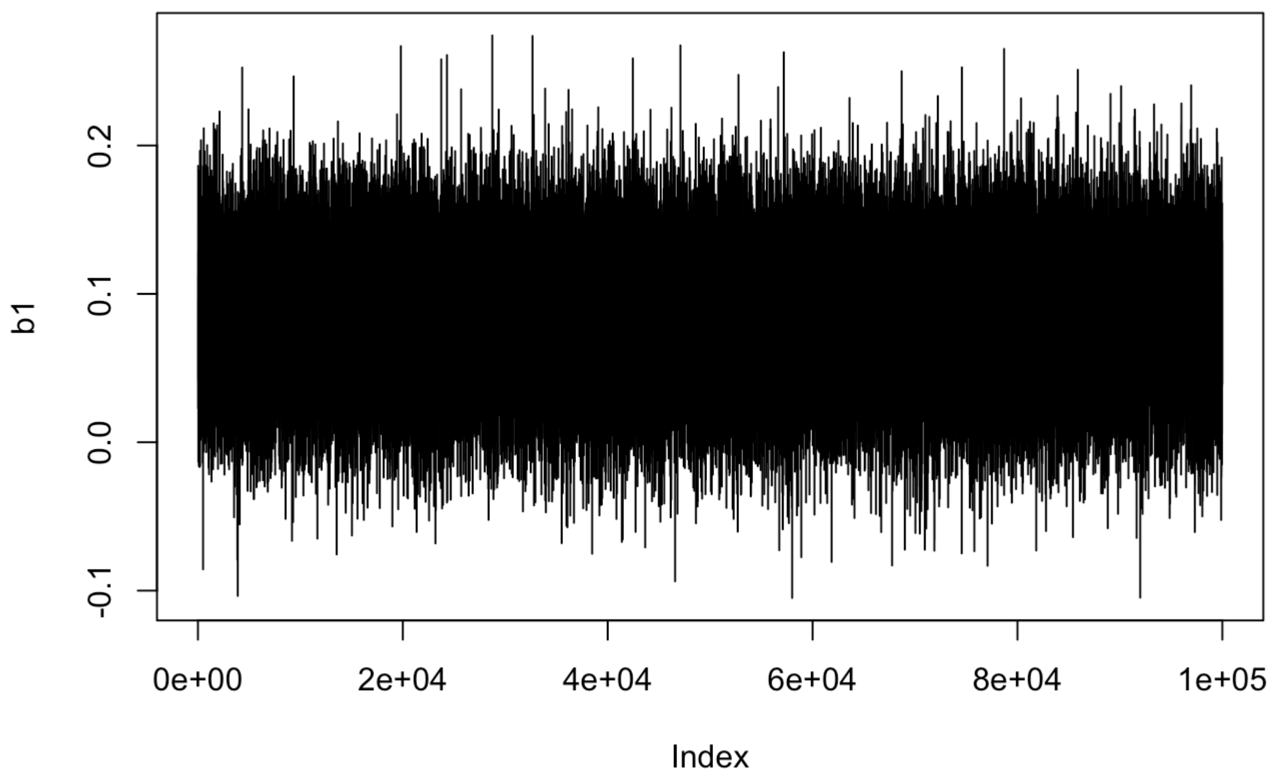
```
age <- c(2.5, 7.5, 12.5, 17.5, 22.5, 27.5)
leaks <- c(2, 1, 4, 4, 2, 0)
t <- c(1052.0, 982.5, 756.9, 442.4, 230.9, 43.9)
n <- length(leaks)

#plot(leaks, age)
#plot(leaks, t)
#plot(leaks, age*t)

LeakModel <- "model {
  for(i in 1:n){
    leaks[i] ~ dpois(lambda[i]*t[i])
    log(lambda[i]) <- beta[1] + beta[2] * age[i]
  }
  beta[1] ~ dnorm(0,1/100)
  beta[2] ~ dnorm(0,1/100)
}
"
#beta[1] & beta[2] are independent priors

LeakModel.sim <- jags(
  data=c('age','leaks','t', 'n'),
  parameters.to.save=c('beta'),
  model.file=textConnection(LeakModel),
  n.iter=22000,
  n.burnin=2000,
  n.chains=5,
  n.thin=1
)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 6
##    Unobserved stochastic nodes: 2
##    Total graph size: 49
##
## Initializing model
```

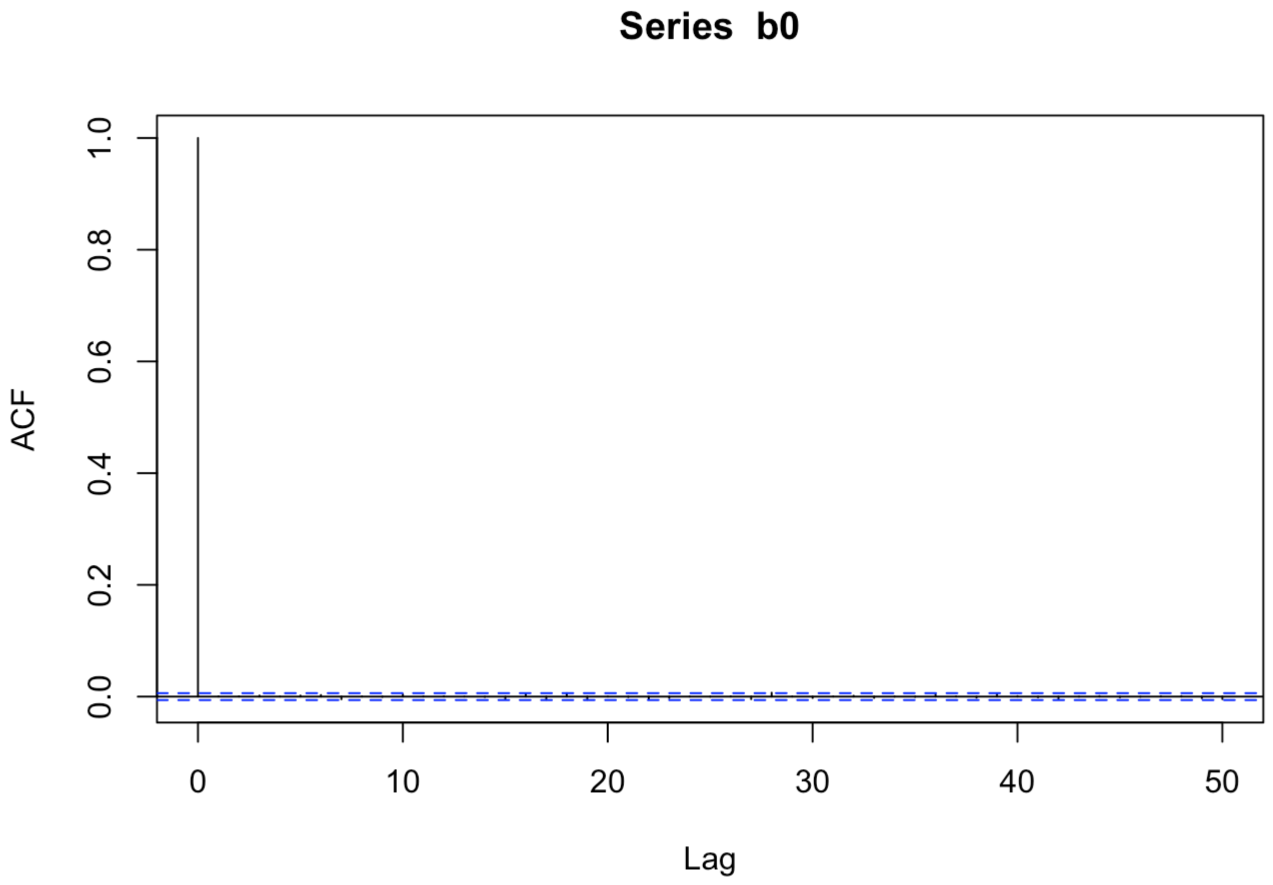
```
# 100,000 MCMC samples obtained

#head(LeakModel.sim$BUGSoutput$sims.matrix)

b0 <- LeakModel.sim$BUGSoutput$sims.matrix[,1]
b1 <- LeakModel.sim$BUGSoutput$sims.matrix[,2]

plot(b0,type="l")
```



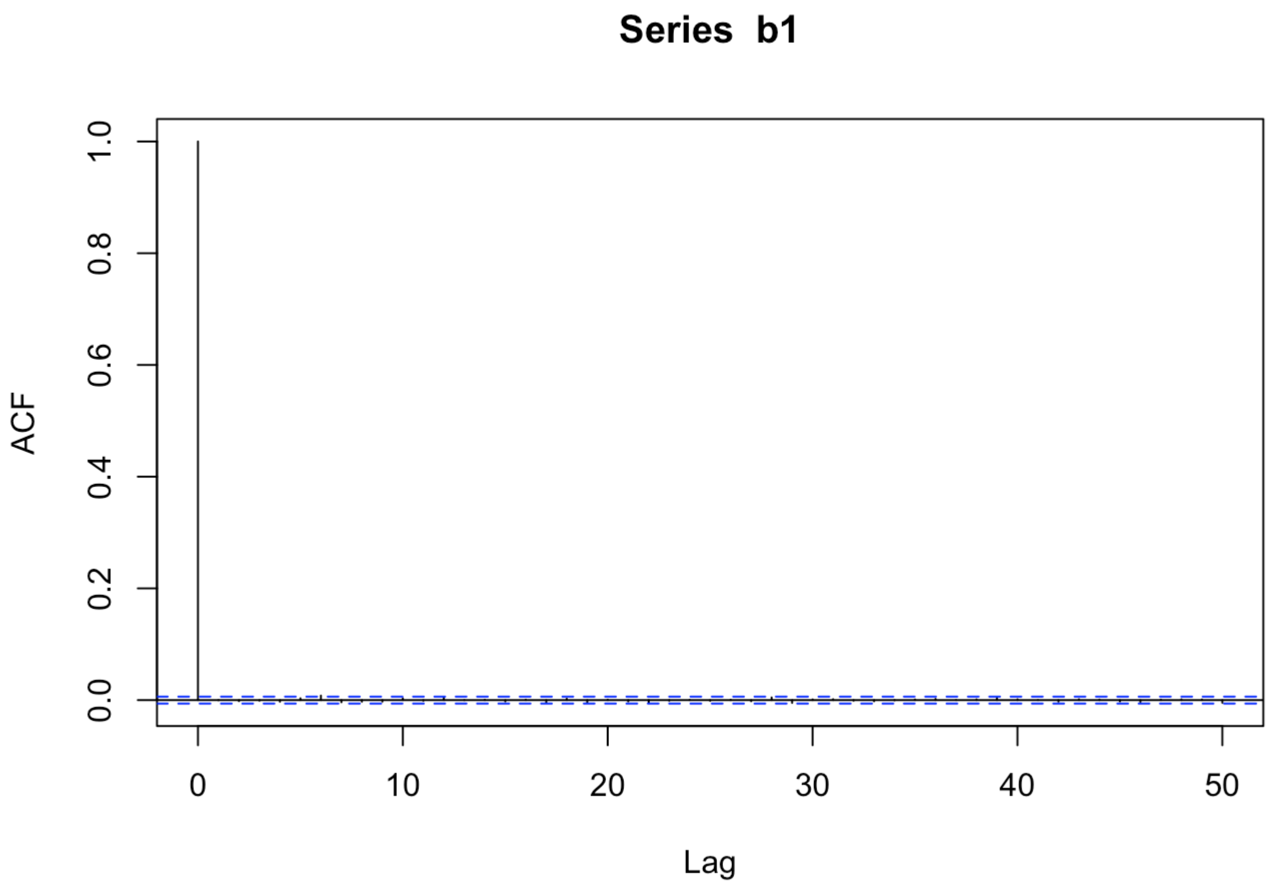
```
plot(b1,type="l")
```



```
acf(b0)
```



```
acf(b1)
```



```
gelman.diag(LeakModel.sim$BUGSoutput)
```

```
## Potential scale reduction factors:
##
##          Point est. Upper C.I.
## beta[1]           1          1
## beta[2]           1          1
## deviance          1          1
##
## Multivariate psrf
##
## 1
```

```
effectiveSize(LeakModel.sim)
```

```
##   beta[1]   beta[2]  deviance
##  7417.693  7593.643 16466.997
```

```
LeakModel.sim$BUGSoutput$DIC
```

```
## [1] 21.18689
```

```r
#######################
# Check Model Fit w/ the GoF Test
y <- leaks

# Calculating the fitted quantiles for each posterior model
# and adjusting for a discrete model
GoF <- matrix(NA,ncol=length(age),nrow=length(b0))
for (i in 1:length(b0)) {
  for (j in 1:length(age)) {
    vals <- ppois(c(y[j]-1,y[j]),exp(b0[i]+b1[i]*(age[j])))
    GoF[i,j] <- runif(1,vals[1],vals[2])
  }
}

# Function requires fitted quantiles and returns a p-value
GoF_Test <- function(fitted_quantiles) {
  n <- length(fitted_quantiles)
  K <- round((n)^(0.4))
  mK <- table(cut(fitted_quantiles,(0:K)/K))
  np <- n/K
  RB <- sum(((mK-np)^2)/np)
  return(1-pchisq(RB,K-1))
}

# Calculating the p-values for each posterior model
GoF_Summary <- apply(GoF,1,GoF_Test)

# Histogram of posterior model p-values
#hist(GoF_Summary,xlim=c(0,1))

# Percent of posterior models with p-value less than 0.05
mean(GoF_Summary < 0.05)
```

```
## [1] 0.49079
```

```r
q1prob <- mean(b1>0)
```

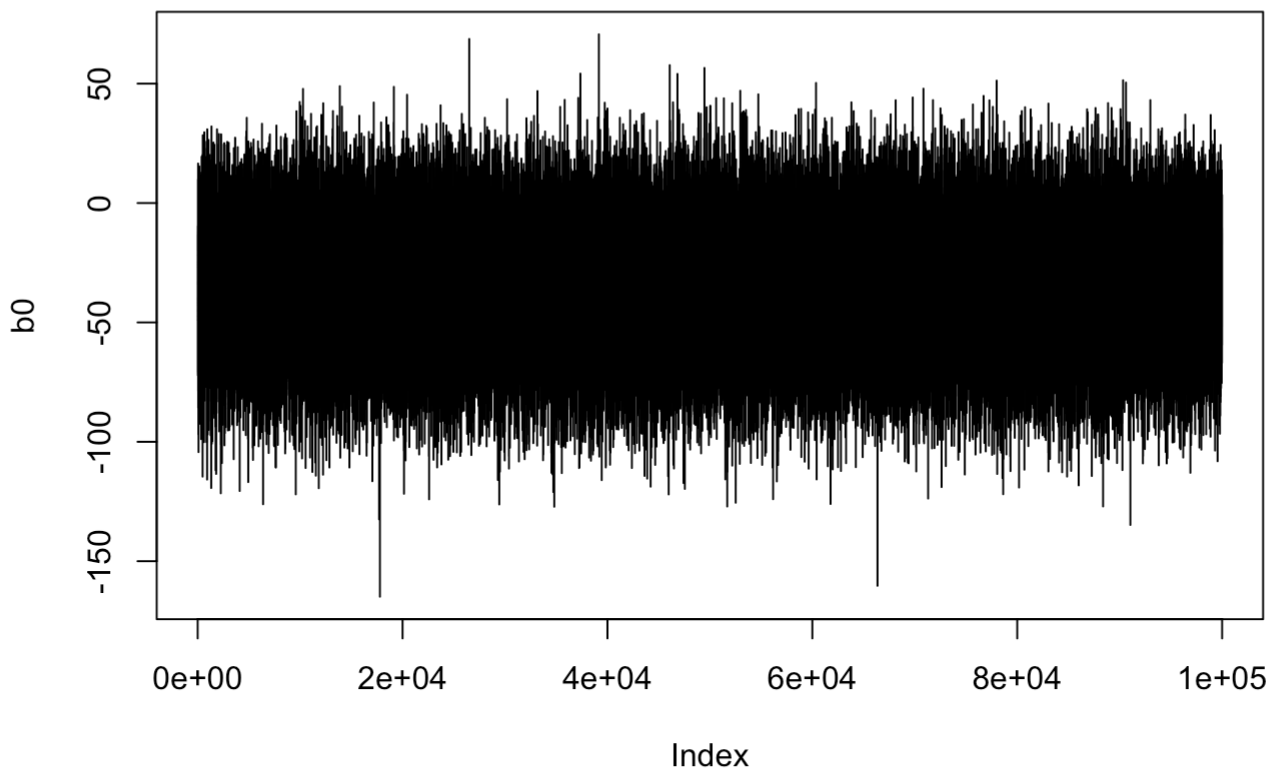
The model does not fit the data well as seen from the Chi-square Goodness-of-Fit test. What is P(β1 > 0) in the following model? 0.98264

Question 2

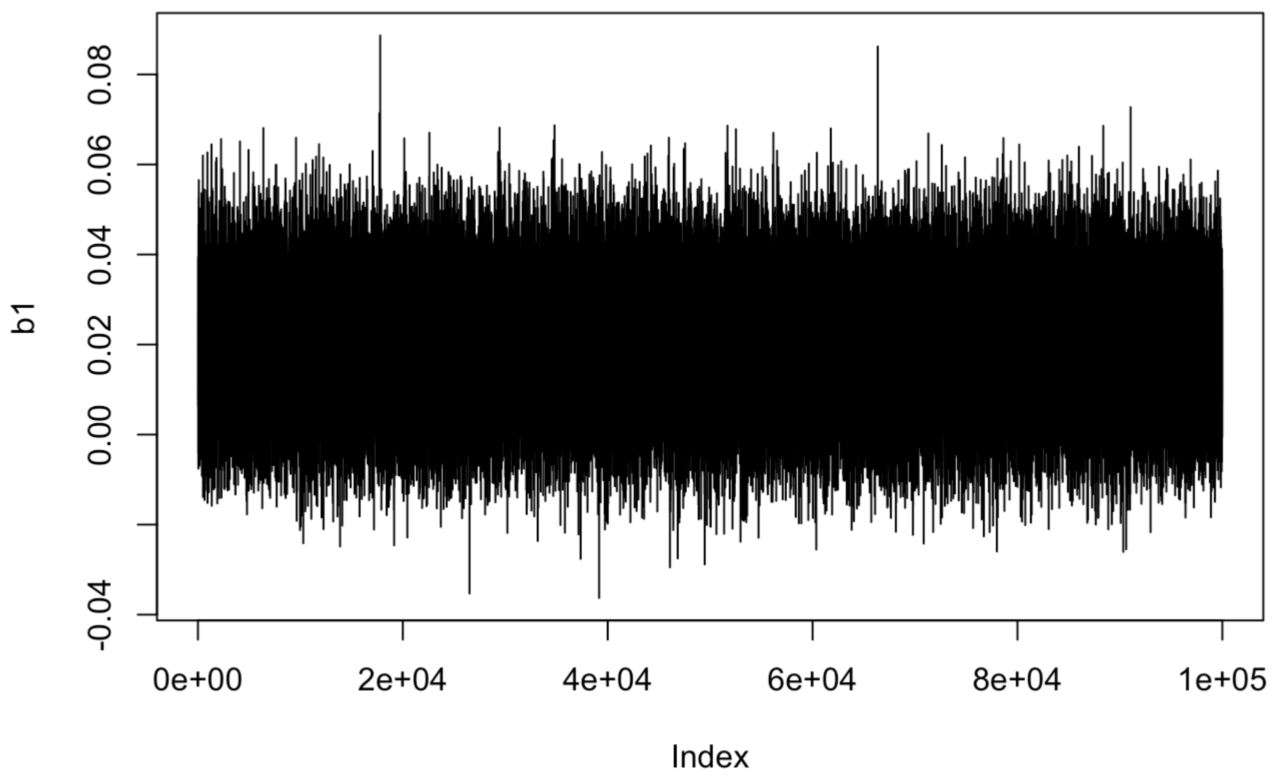
```
HKdata <- read.table("HorseKicks.txt",header=T)

horse <- HKdata %>%
  mutate(deaths = rowSums(select(., GC:C15)), year = Year)

#plot(horse$year, horse$deaths)

#x <- seq(0, 20, length = 1001)
#curve(dexp(x, rate=1/5), from=0, to=20) #mean of 5
#curve(dexp(x, rate=1/15), from=0, to=20) #mean of 15

#q <- matrix(c(1,1,1876,1892), ncol=2)
#q
#solve(q)

n <- length(horse$deaths)
time <- horse$year
y <- horse$deaths

HorseModel <- "model {
  for(i in 1:n){
    y[i] ~ dpois(lambda[i])
    log(lambda[i]) <- beta[1] + beta[2] * time[i]
  }

  beta[1] = (118.2500)*log(lam76) - (117.2500)*log(lam92)
  beta[2] = -(0.0625)*log(lam76) + (0.0625)*log(lam92)
  lam76 ~ dexp(1/5)
  lam92 ~ dexp(1/15)

}
"

HorseModel.sim <- jags(
  data=c('y','time','n'),
  parameters.to.save=c('beta'),
  model.file=textConnection(HorseModel),
  n.iter=22000,
  n.burnin=2000,
  n.chains=5,
  n.thin=1
)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 20
##    Unobserved stochastic nodes: 2
##    Total graph size: 120
##
## Initializing model
```

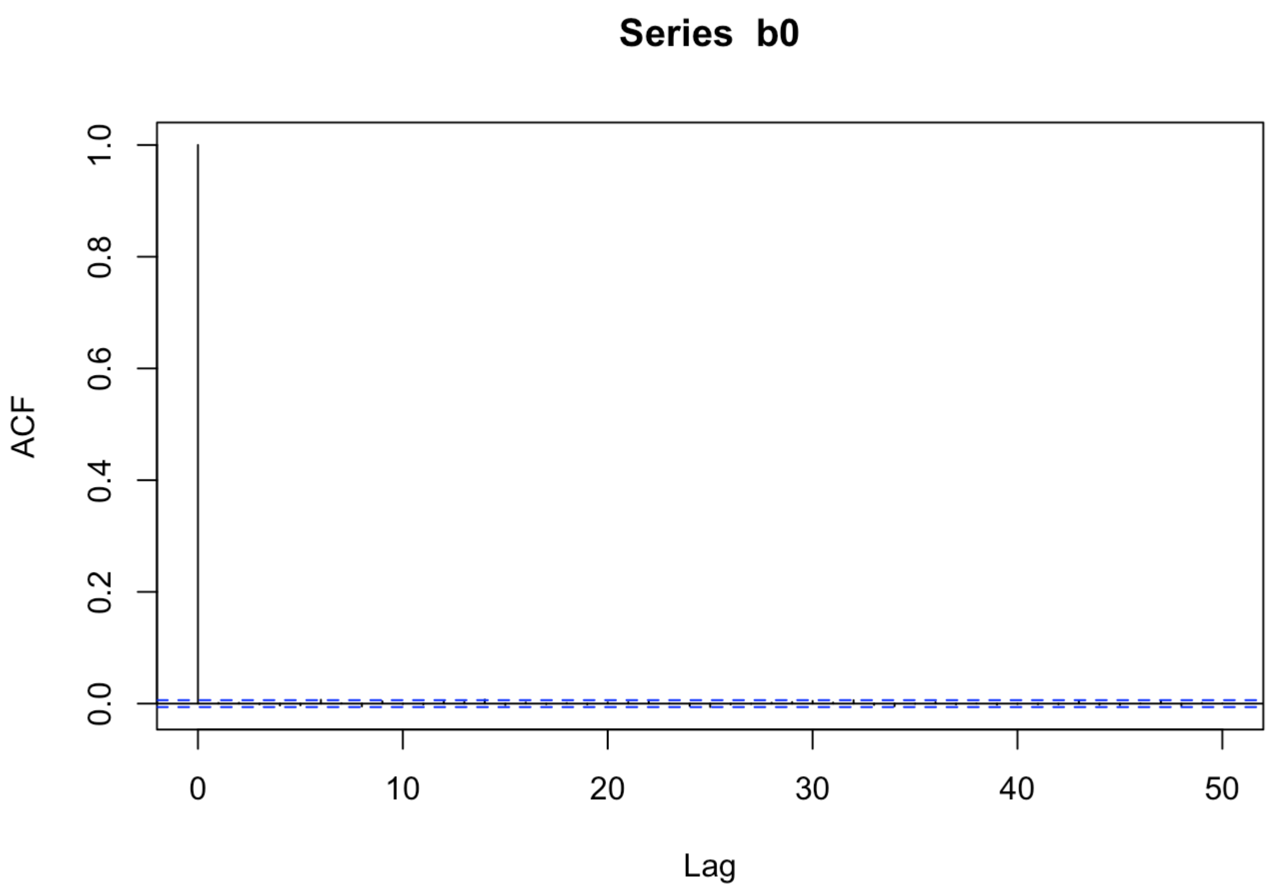
```
# 100,000 MCMC samples obtained
#head(HorseModel.sim$BUGSoutput$sims.matrix)

b0 <- HorseModel.sim$BUGSoutput$sims.matrix[,1]
b1 <- HorseModel.sim$BUGSoutput$sims.matrix[,2]

plot(b0,type="l")
```



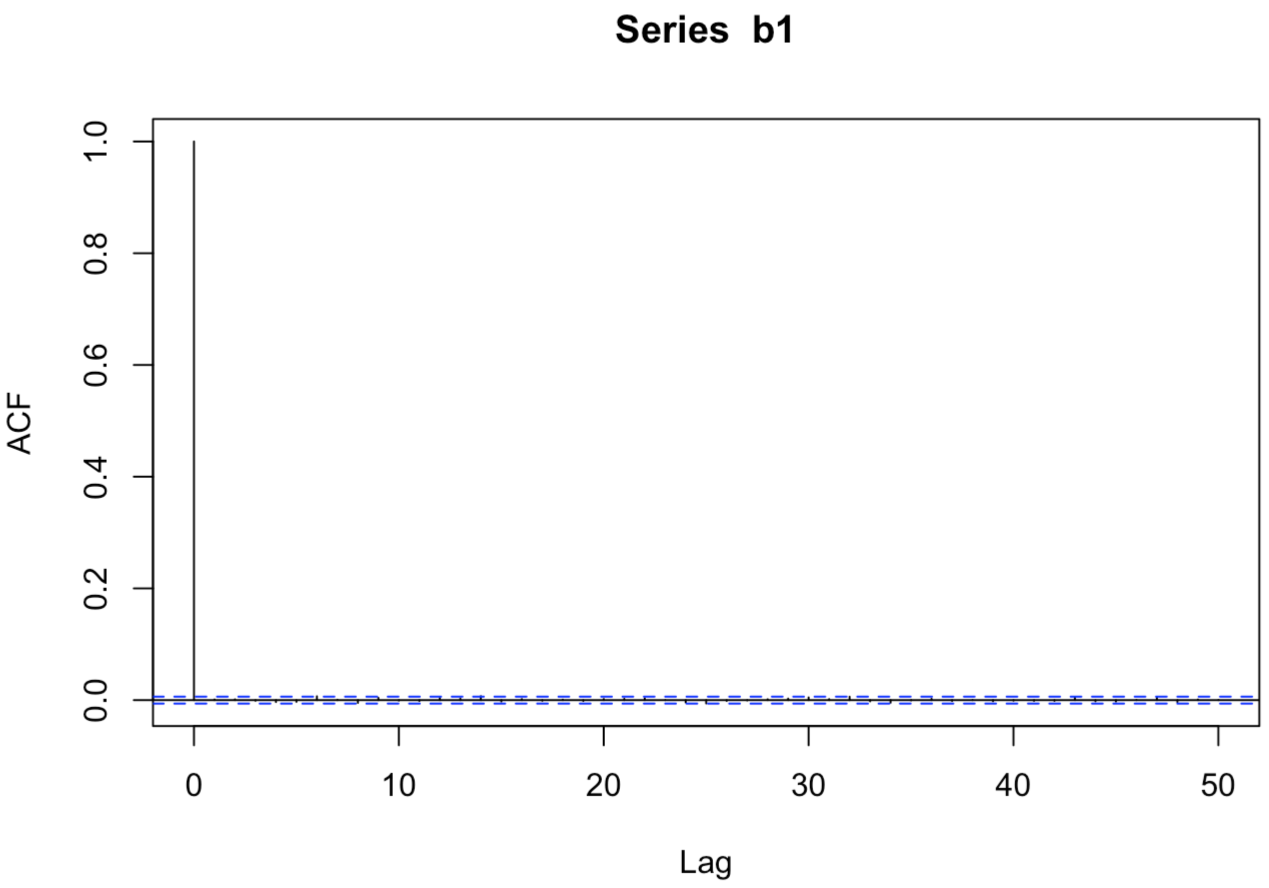
```
plot(b1,type="l")
```



```
acf(b0)
```



```r
acf(b1)
```



```r
gelman.diag(HorseModel.sim$BUGSoutput)
```

```
## Potential scale reduction factors:
##
##          Point est. Upper C.I.
## beta[1]           1          1
## beta[2]           1          1
## deviance          1          1
##
## Multivariate psrf
##
## 1
```

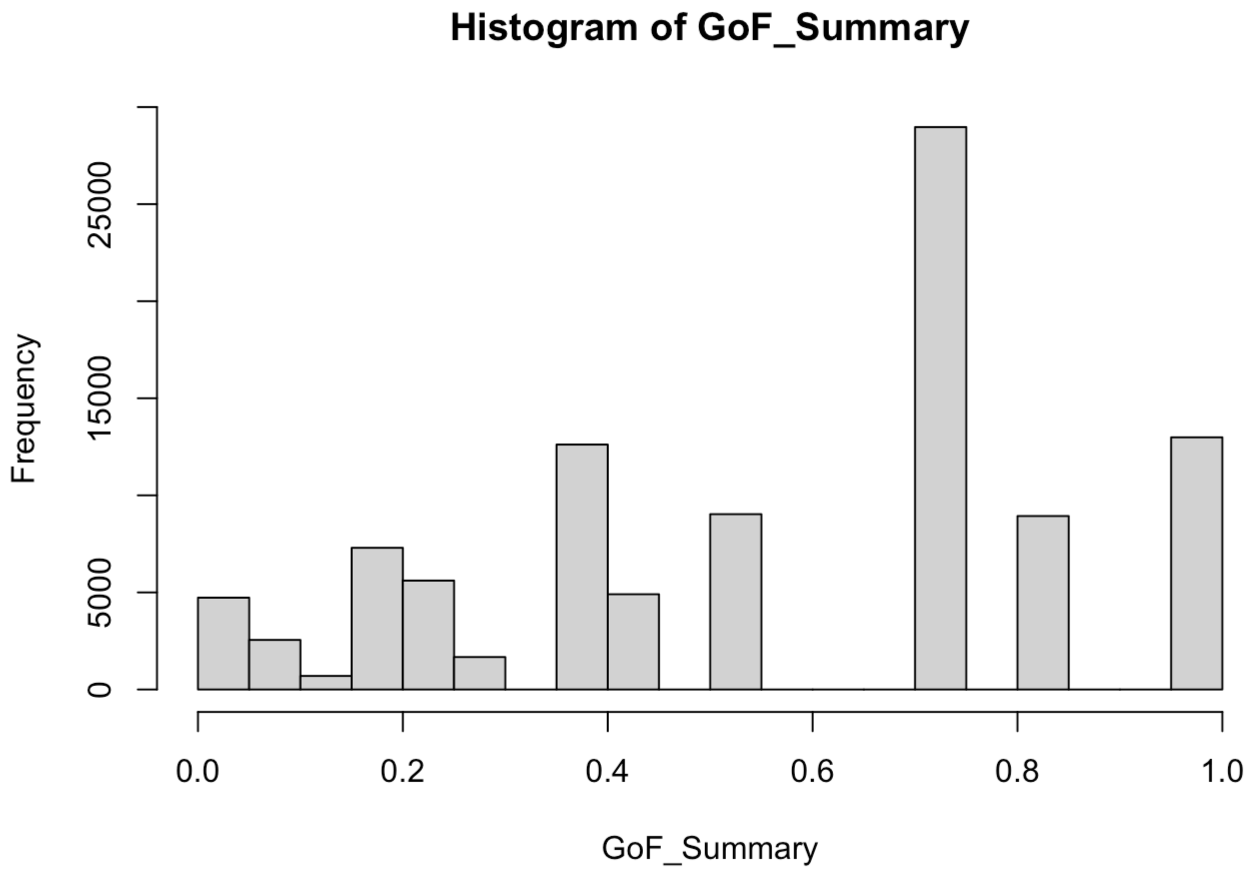
```r
HorseModel.sim$BUGSoutput$DIC
```

```
## [1] 120.8834
```

```r
effectiveSize(HorseModel.sim)
```

```
##  beta[1]  beta[2] deviance
## 43799.65 43791.95 42538.03
```

```r
# Calculating the fitted quantiles for each posterior model
# and adjusting for a discrete model
GoF <- matrix(NA,ncol=length(time),nrow=length(b0))
for (i in 1:length(b0)) {
  for (j in 1:length(time)) {
    vals <- ppois(c(y[j]-1,y[j]),exp(b0[i]+b1[i]*(time[j])))
    GoF[i,j] <- runif(1,vals[1],vals[2])
  }
}

# Function requires fitted quantiles and returns a p-value
GoF_Test <- function(fitted_quantiles) {
  n <- length(fitted_quantiles)
  K <- round((n)^(0.4))
  mK <- table(cut(fitted_quantiles,(0:K)/K))
  np <- n/K
  RB <- sum(((mK-np)^2)/np)
  return(1-pchisq(RB,K-1))
}

# Calculating the p-values for each posterior model
GoF_Summary <- apply(GoF,1,GoF_Test)

# Histogram of posterior model p-values
hist(GoF_Summary,xlim=c(0,1))
```



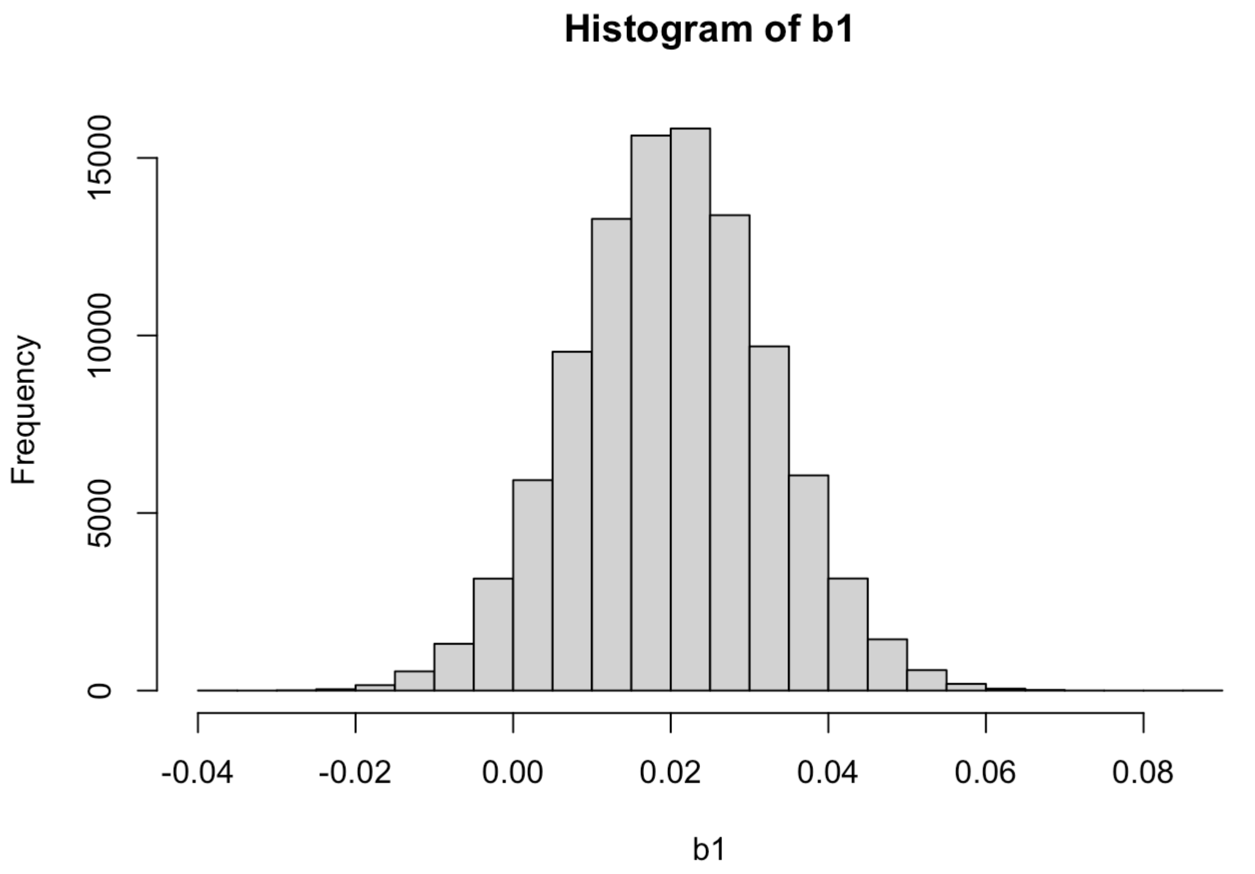
```
# Percent of posterior models with p-value less than 0.05
mean(GoF_Summary < 0.05)
```

```
## [1] 0.0473
```

```
mean(b1)
```

```
## [1] 0.02011894
```

```
hist(b1)
```



```
#lam76 <- rexp(100000,1/5)
#lam92 <- rexp(100000,1/15)
#b0prior <- (118.2500)*log(lam76) - (117.2500)*log(lam92)
#b1prior <- -(0.0625)*log(lam76) + (0.0625)*log(lam92)
#plot(b0prior,b1prior,xlim=c(-15,7),ylim=c(-.03,.03),pch='.',ylab='b1',xlab='b0')
#par(new=T)
#plot(HorseModel.sim$BUGSoutput$sims.matrix[,1],HorseModel.sim$BUGSoutput$sims.matrix[,2],
#     xlim=c(-15,20),ylim=c(-.03,.03),col=4,pch='.',ylab='',xlab='')

lambda1880 <- exp(b0+b1*(1880))
PPD1880 <- rpois(length(lambda1880),lambda1880)
quantile(PPD1880,c(0.025,0.975))
```

```
##  2.5% 97.5%
##     3    15
```

```
lambda1890 <- exp(b0+b1*(1890))
PPD1890 <- rpois(length(lambda1890),lambda1890)
quantile(PPD1890,c(0.025,0.975))
```

```
##  2.5% 97.5%
##     5    18
```

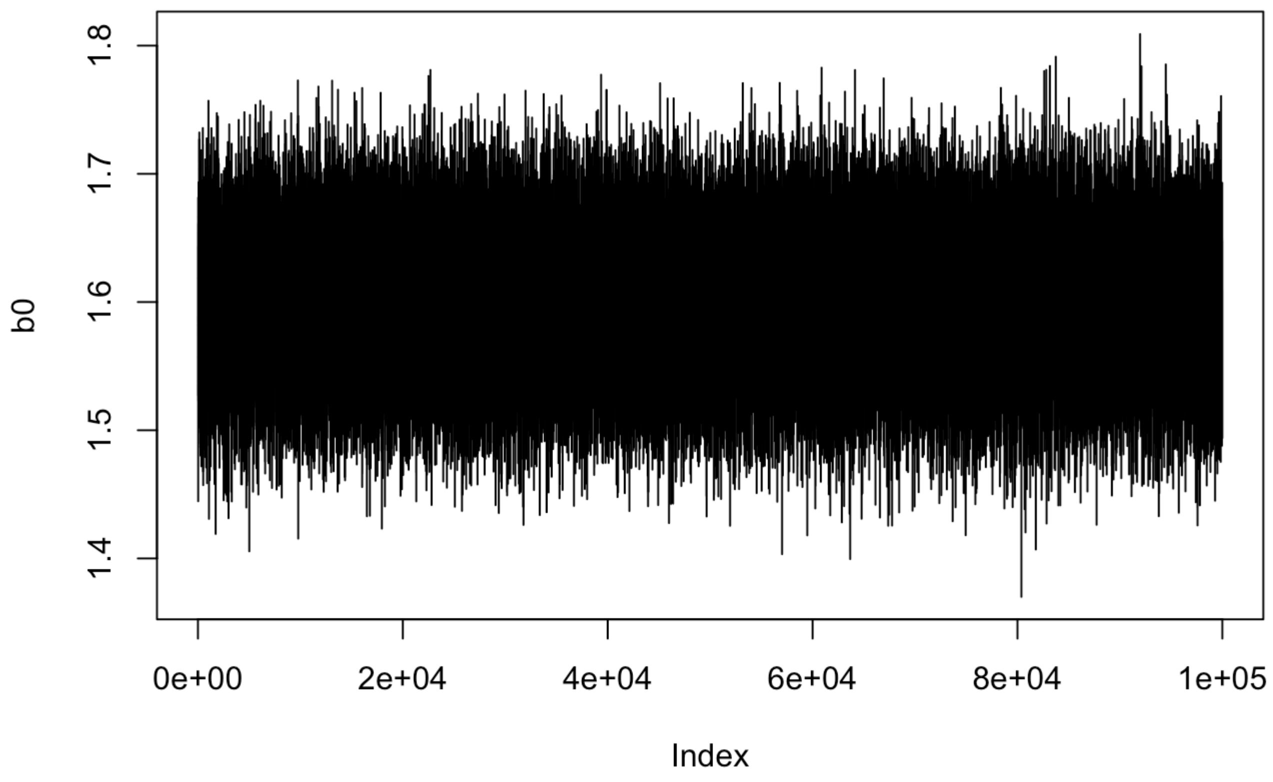
Time does affect the rate of death by horse kicks. Our model fits the data well, as seen from the Chi-square Goodness-of-Fit test, and our b1 value is positive. This means that our slope is increasing over time. Thus, there are more deaths per year as time goes on.

Question 3

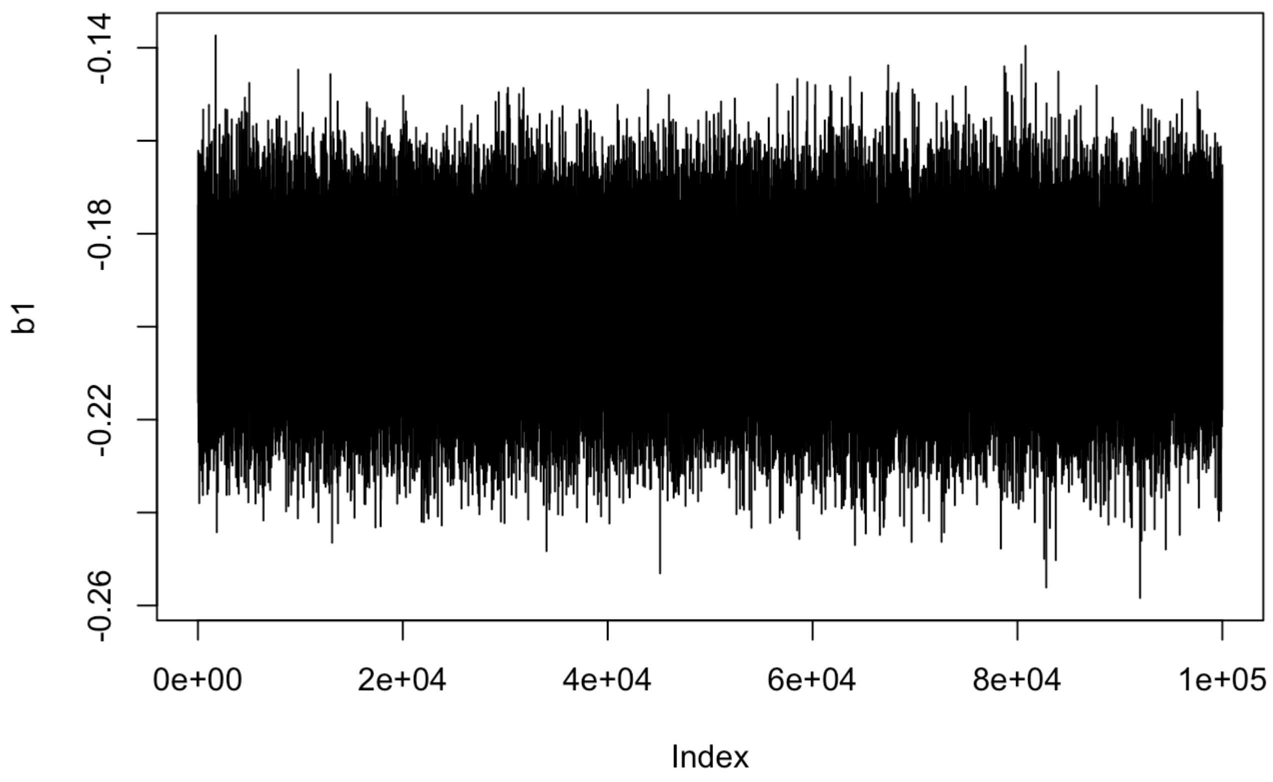
```
gauge <- read.table("table78.txt",header=T)
#head(gauge)

#plot(gauge$Gauge, gauge$Strength)
#plot(log(gauge$Gauge), gauge$Strength)

#z <- matrix(c(1,1,1.609438,5.579730), ncol=2)
#z
#solve(z)

logGauge <- log(gauge$Gauge)
n <- length(gauge$Strength)
y <- gauge$Strength


GaugeModel <- "model {
  for(i in 1:n){
    y[i] ~ dlnorm(mu[i],1/(sigma)^2)
    mu[i] <- beta0 + beta1 * logGauge[i]
  }

  sigma ~ dexp(1/2)
  mu1.6 ~ dnorm(1.7128,0.2)
  mu5.5 ~ dnorm(3.5272,0.2)
  beta0 = (1.4053702)*mu1.6 - (0.4053702)*mu5.5
  beta1 = -(0.2518706)*mu1.6 + (0.2518706)*mu5.5

}
"

GaugeModel.sim <- jags(
  data=c('y', 'n', 'logGauge'),
  parameters.to.save=c('beta0','beta1','sigma'),
  model.file=textConnection(GaugeModel),
  n.iter=22000,
  n.burnin=2000,
  n.chains=5,
  n.thin=1
)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 214
##    Unobserved stochastic nodes: 3
##    Total graph size: 458
##
## Initializing model
```

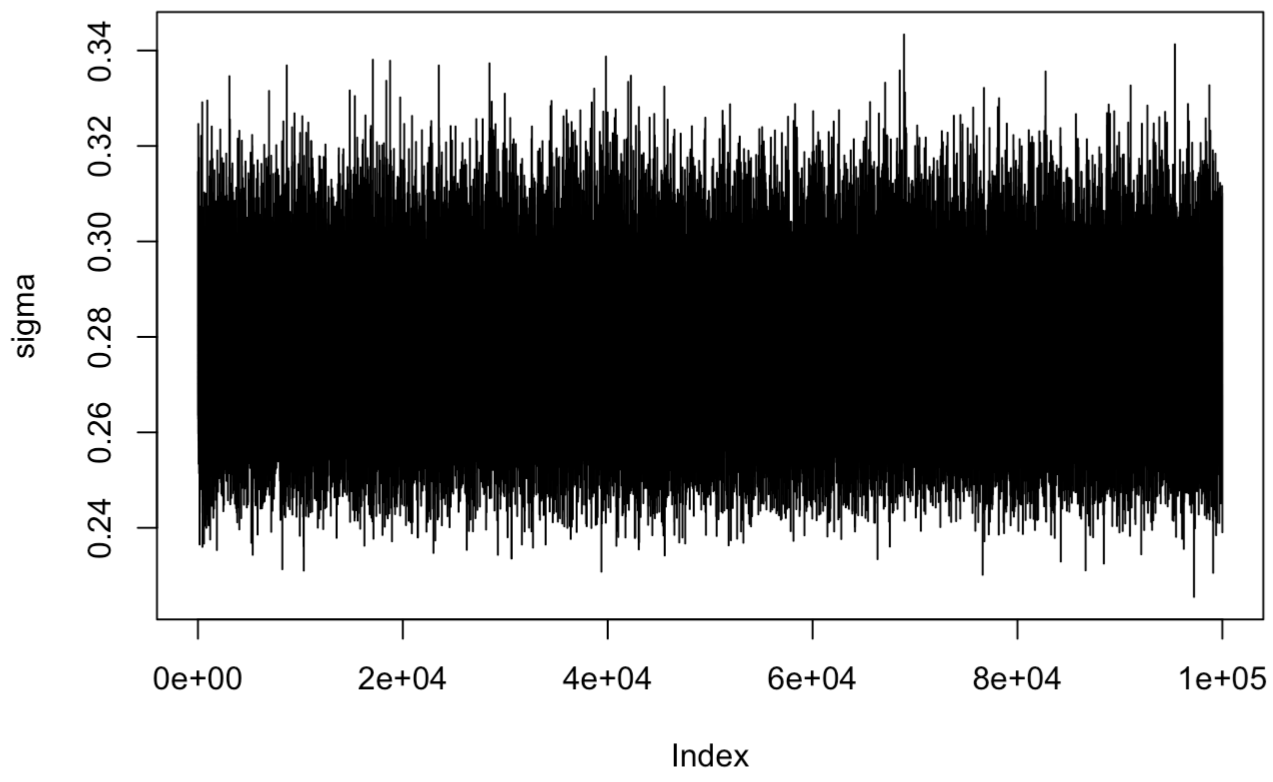
```
#head(GaugeModel.sim$BUGSoutput$sims.matrix)

b0 <- GaugeModel.sim$BUGSoutput$sims.matrix[,1]
b1 <- GaugeModel.sim$BUGSoutput$sims.matrix[,2]
sigma <- GaugeModel.sim$BUGSoutput$sims.matrix[,4]

plot(b0,type="l")
```



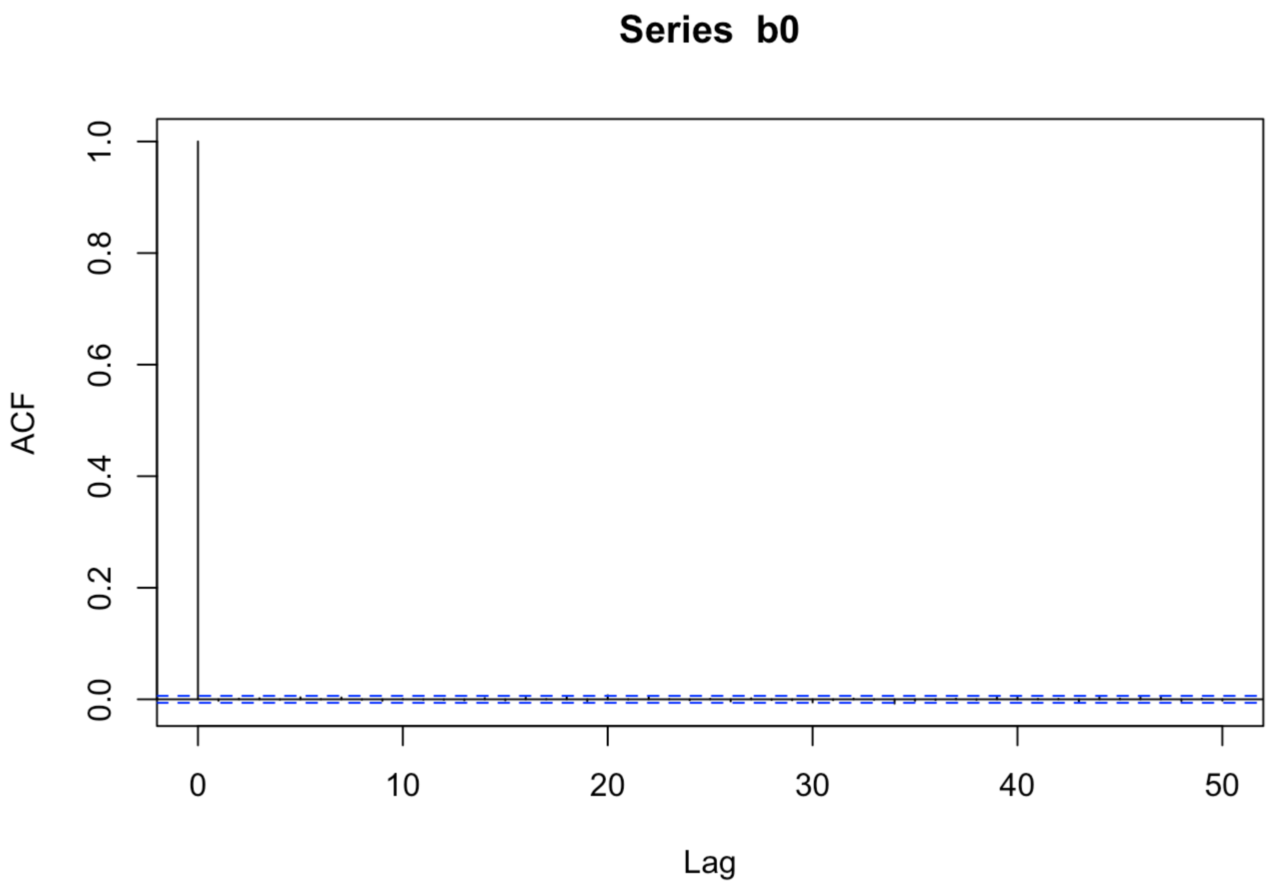
```
plot(b1,type="l")
```



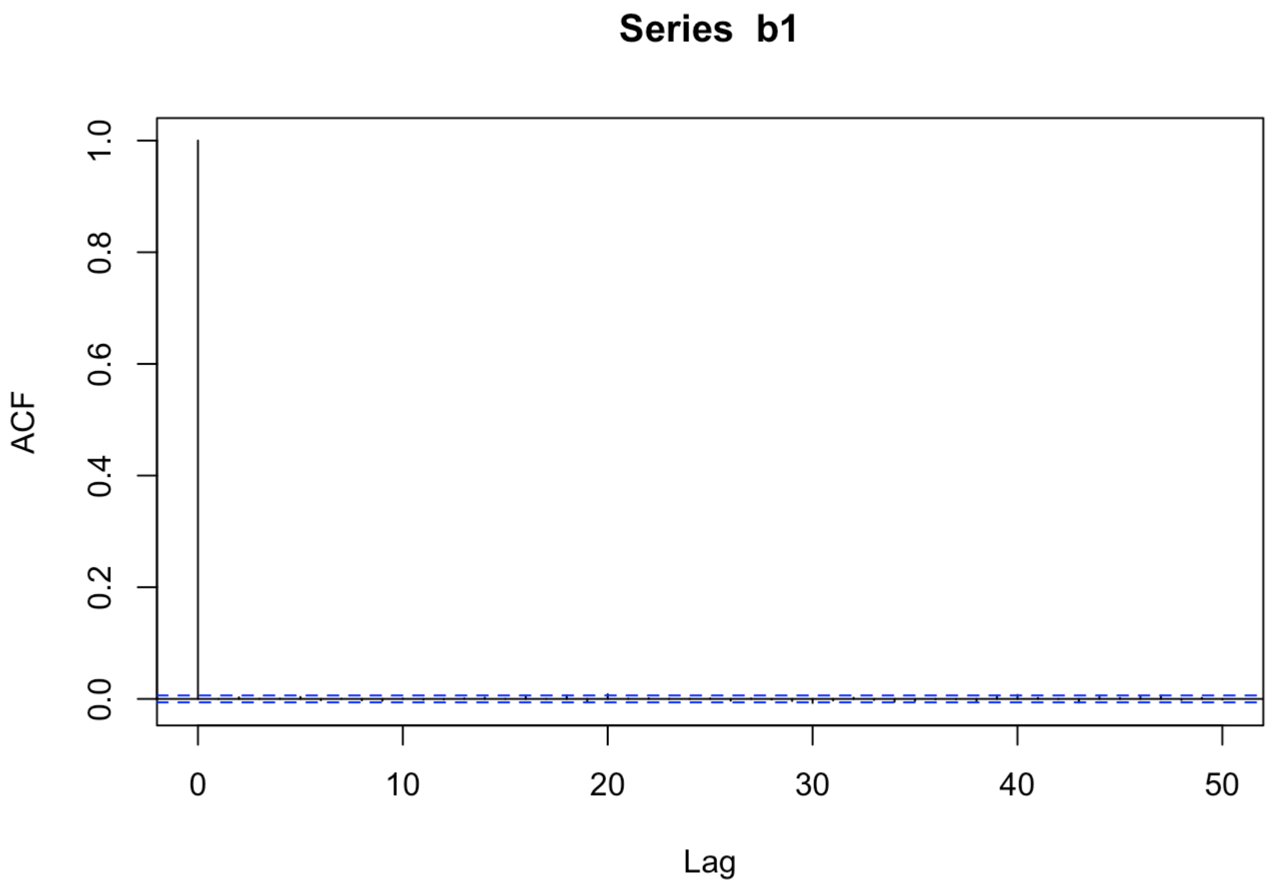
```
plot(sigma,type="l")
```



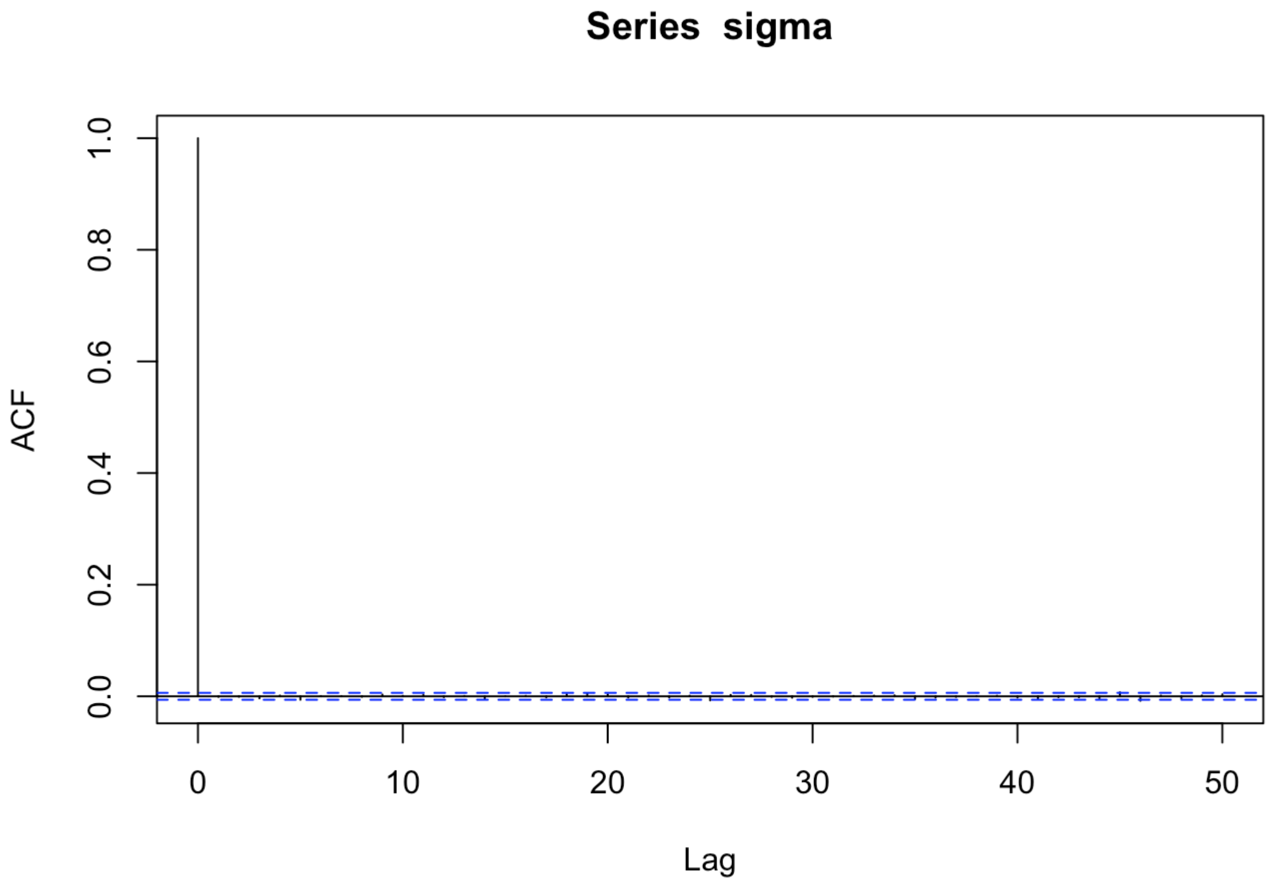
```
acf(b0)
```



```
acf(b1)
```



```
acf(sigma)
```



```
gelman.diag(GaugeModel.sim$BUGSoutput)
```

```
## Potential scale reduction factors:
##
##          Point est. Upper C.I.
## beta0             1          1
## beta1             1          1
## deviance          1          1
## sigma             1          1
##
## Multivariate psrf
##
## 1
```

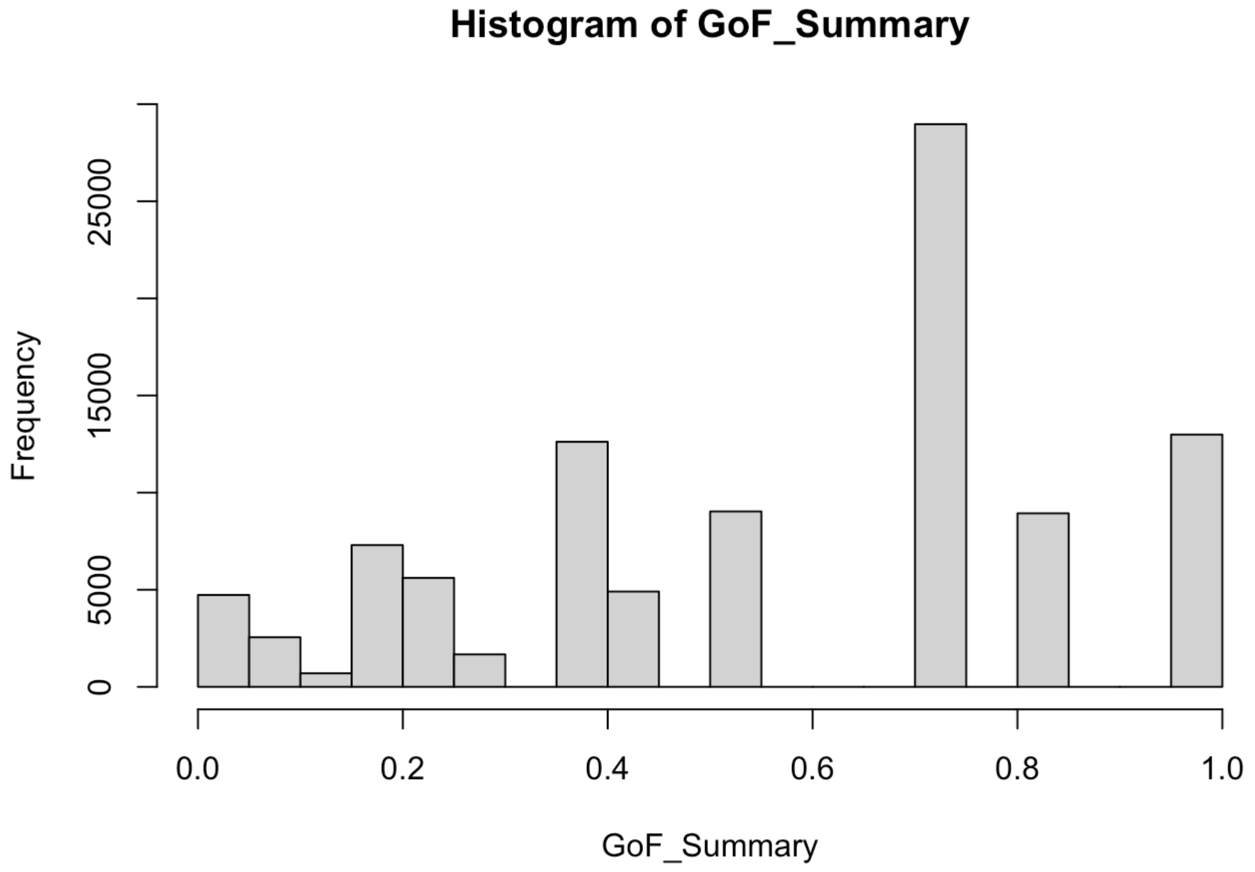
```
GaugeModel.sim$BUGSoutput$DIC
```

```
## [1] 471.6606
```

```
effectiveSize(GaugeModel.sim)
```

```
##    beta0    beta1 deviance    sigma
## 44127.74 42289.71 40160.25 57195.98
```

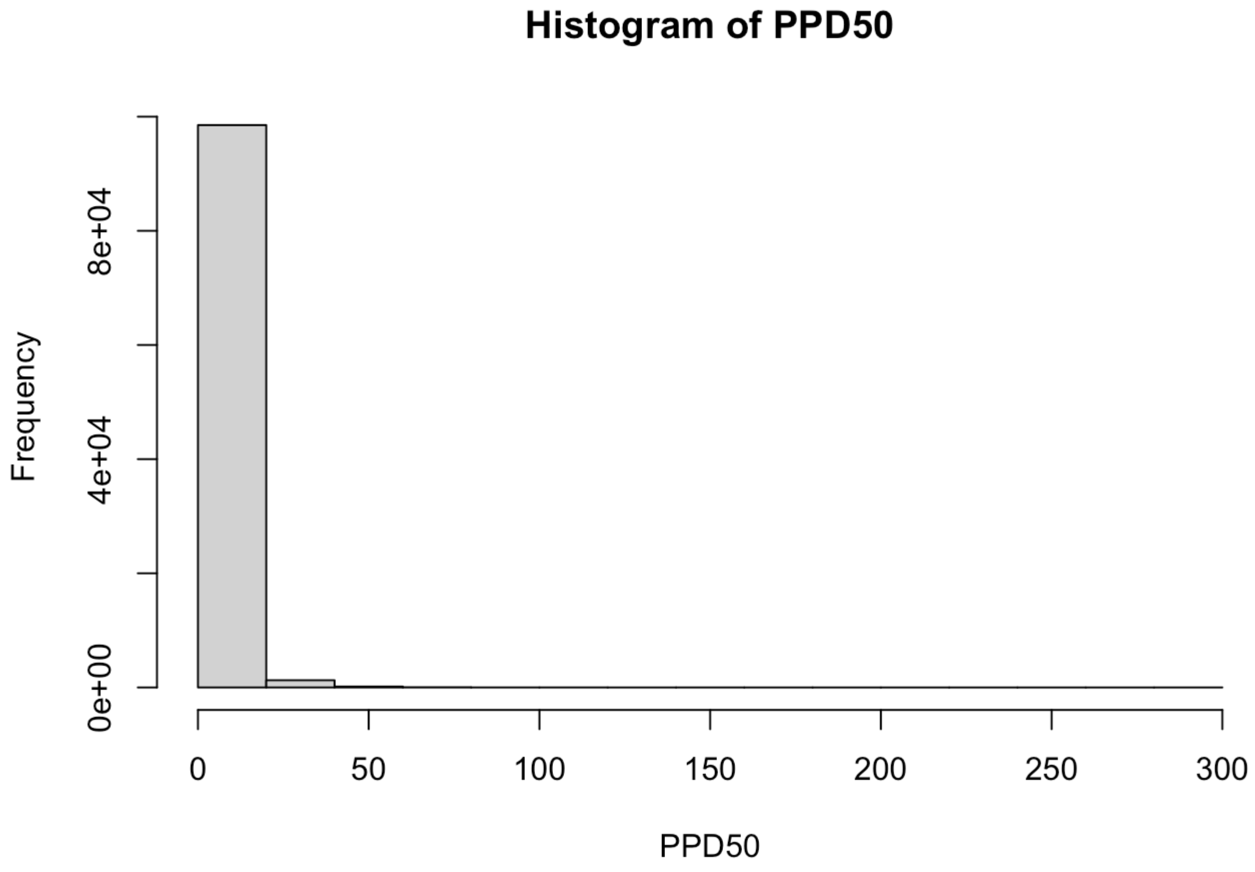
```
GoF <- matrix(NA,ncol=n,nrow=length(b0))
for (i in 1:length(b0)) {
  for (j in 1:n) {
    GoF[i,j] <- plnorm(y[j],b0[i]+b1[i]*logGauge[j],sd=sigma[i])
  }
}

GoF_Test <- function(fitted_quantiles) {
  n <- length(fitted_quantiles)
  K <- round((n)^(0.4))
  mK <- table(cut(fitted_quantiles,(0:K)/K))
  np <- n/K
  RB <- sum(((mK-np)^2)/np)
  return(1-pchisq(RB,K-1))
}

GoF <- apply(GoF,1,GoF_Test)

hist(GoF_Summary,xlim=c(0,1))
```



```
# Percent of posterior models with p-value less than 0.05
mean(GoF_Summary < 0.05)
```

```
## [1] 0.0473
```

```
# PPD for gauge length 50mm
mu50 <- b0+b1*(log(50))
PPD50 <- rlnorm(length(mu50),mu50)

hist(PPD50)
```



```
#summary(PPD50)
estimate <- mean(PPD50)
#quantile(PPD50,c(0.025,0.975))
less.than.5 <- quantile(PPD50,c(0.05,1))
```

Model fits quite well, as seen from the Chi-square Goodness-of-Fit diagnostics.

estimate: 3.7621285

uncertainty such that less than 5% of the population has a smaller value: (0.4415513, 292.3085559)