HW5

Isaac Fan

3/19/2021

```
library(coda)
library(readr)
library(actuar)
```

```
##
## Attaching package: 'actuar'

## The following object is masked from 'package:grDevices':
##
##     cm
```

```
library(invgamma)
```

```
##
## Attaching package: 'invgamma'

## The following objects are masked from 'package:actuar':
##
##     dinvexp, dinvgamma, pinvexp, pinvgamma, qinvexp, qinvgamma,
##     rinvexp, rinvgamma
```

```
library(LearnBayes)
library(EnvStats)
```

```
##
## Attaching package: 'EnvStats'

## The following objects are masked from 'package:actuar':
##
##     dpareto, ppareto, qpareto, rpareto

## The following objects are masked from 'package:stats':
##
##     predict, predict.lm

## The following object is masked from 'package:base':
##
##     print.default
```

```
library(purrr)
library(ggplot2)
claims_dat <- read_csv("~/claims.dat.txt")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Claims = col_double()
## )
```

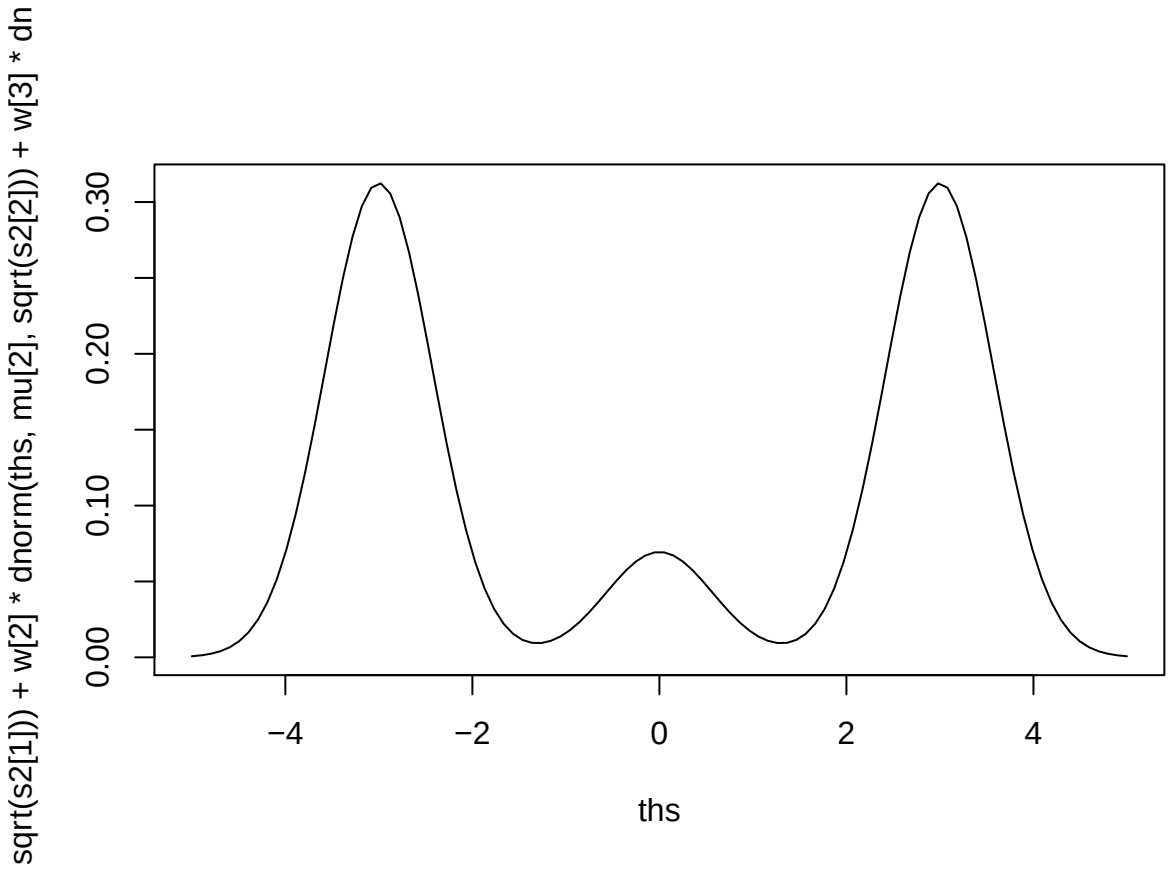
```
glucose.dat <- read_csv("~/glucose.dat.txt")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Glucose = col_double()
## )
```

Question 1

Part C

```
#### Mixture normal parameters
mu<-c(-3,0,3) # mean vector
s2<-c(.33,.33,.33) # variance vector
w<-c(.45,.1,.45) # weights

ths<-seq(-5,5,length=100) # visualize density
plot(ths, w[1]*dnorm(ths,mu[1],sqrt(s2[1])) +
       w[2]*dnorm(ths,mu[2],sqrt(s2[2])) +
       w[3]*dnorm(ths,mu[3],sqrt(s2[3])) ,type="l")
```




```
#### MC Sampling
set.seed(1)
S<-10000
d<-sample(1:3,S, prob=w,replace=TRUE)
th<-rnorm(S,mu[d],sqrt(s2[d]))
THD.MC<-cbind(th,d)

#### MC Confidence Interval
c(mean(pnorm(th < 3)) - 1.96 * sd(pnorm(th < 3)) / sqrt(10000), mean(pnorm(th < 3)) + 1.96 * sd(pnorm(t
```

```
## [1] 0.7653799 0.7708727
```

```
#### MCMC: Gibbs sampling
th<-0 # initial value for X
S<-10000
THD.MCMC<-matrix(NA,nrow=S,ncol=2)
set.seed(1)
for(s in 1:S) {
  d<-sample(1:3 ,1,prob= w*dnorm(th,mu,sqrt(s2))) #sampling full conditional d|th
  th<-rnorm(1,mu[d],sqrt(s2[d]) ) #sampling full conditional th|d
  THD.MCMC[s,]<-c(th,d)
}

ess <- effectiveSize(THD.MCMC[,1]) #effective sample size for theta samples
```

```
#### MCMC CI
c(mean(pnorm(THD.MCMC[,1]< 3)) - 1.96 * sd(pnorm(THD.MCMC[,1]<3))
  / sqrt(ess), mean(pnorm(THD.MCMC[,1]< 3)) + 1.96 *
    sd(pnorm(THD.MCMC[,1]<3) / sqrt(ess)))
```

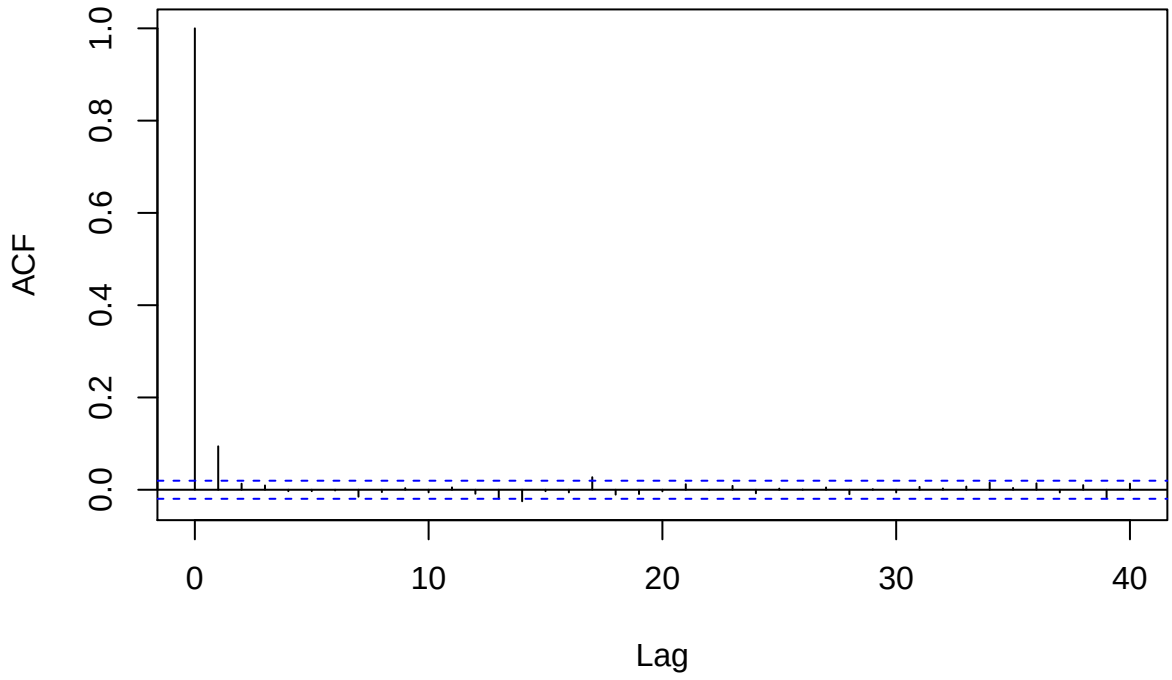
```
##      var1
## 0.7134401 0.8367393
```

Question 2

#Part D

```
set.seed(456)
n<-10000
#### MCMC: Gibbs sampling
  # storage and initialization
m = rep(950, n)
alpha = rep(.5, n)

# sample from the full conditionals
for (j in 2:n) {

  alpha[j] <- rgamma(1, 15 + .01, .01 +
                     sum(log(claims_dat$Claims)) - 15*log(m[j-1]))
  m[j] <- 1/rpareto(1, 1/min(claims_dat$Claims), 15*alpha[j] - 2)
  }
```

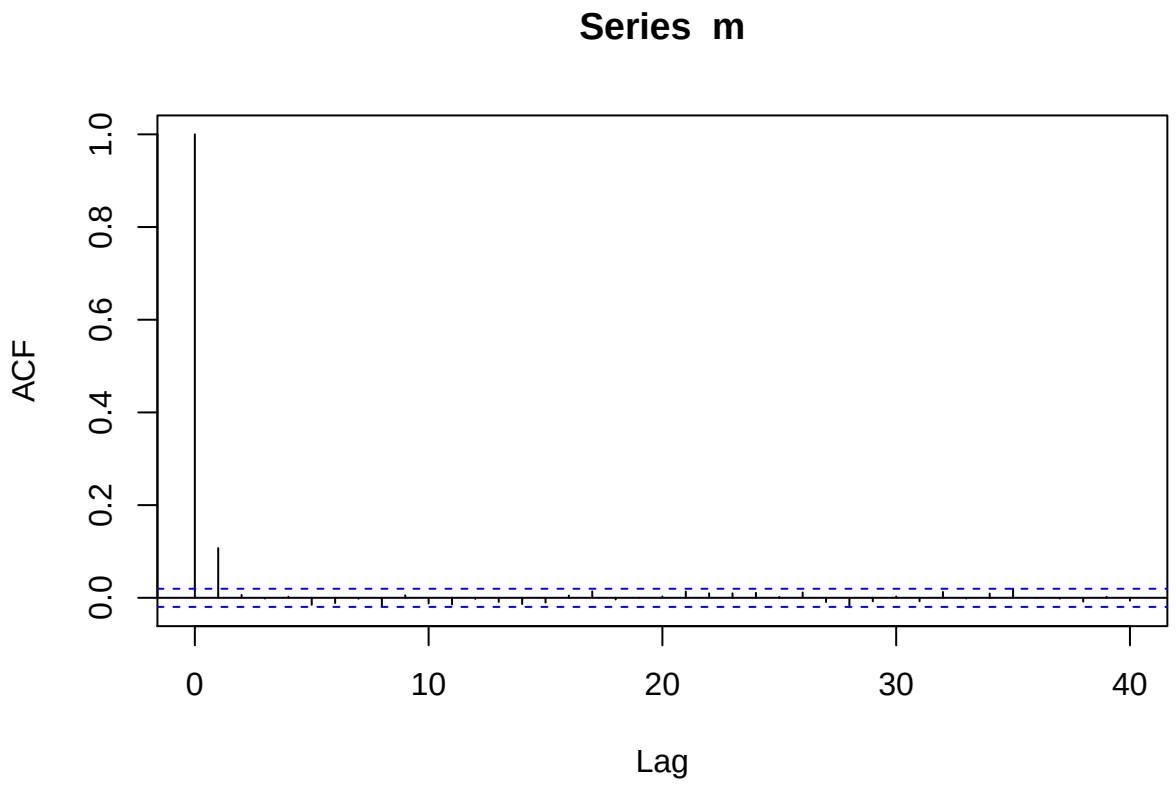
```
acf(alpha)
```




```
effectiveSize(alpha)
```

```
##     var1
## 8278.341
```

```
acf(m)
```




```
effectiveSize(m)
```

```
##     var1
## 8062.727
```

Part E

```
set.seed(2)
total <- rep(0, 10000)
total_costs <- rep(0,1000)
n <- c(rpois(10000, 5))
for (j in 1:10000) {
  #rpareto will not let me sample zero times
  if(n[j] > 0){
    total[j] = sum(rpareto(n[j], m[j],alpha[j]))
    total_costs[j] = total[j]-n[j]*m[j]
  }
  else{
    total[j] = sum(rpareto(1, m[j],alpha[j]))
    total_costs[j] = total[j]-1*m[j]
  }
}
median(total)
```

```
## [1] 21038.16
```

```
median(total_costs)
```

```
## [1] 15410.64
```

```
quantile(total_costs, .75)
```

```
##      75%
## 44481.63
```

Question 3

Part C

```
set.seed(5678)
#prior parameters
a = 1
b = 1
mu = 120
tao = 200
gamma = 1000
v = 10
#create vectors to store outputs
theta1 <- rep(120, 10000)
theta2 <- rep(120, 10000)
sigma1 <- rep(1250, 10000)
sigma2 <- rep(1250, 10000)
#data vector
y <- c(glucose.dat$Glucose)
#generate initial p
p <- rbeta(1, a, b)
p_vec <- rep(.5, 10000)
p_vec[1] <- p

for (i in 2:10000) {
  #create 532 x's for the 532 y's each iteration
  x <- rep(0,532)
  for (j in 2:532) {
    #update bernoulli parameter
    p_star <- p*dnorm(y[j],
          theta2[5],sigma2[i-1])/((1-p)*dnorm(y[j],theta1[i-1],
                sigma1[i-1])+p*dnorm(y[j], theta2[i-1],
                                     sigma2[i-1]))
    x[j] <- rbernoulli(1, p_star)
  }
  #sample p for next iteration
  p <- rbeta(1, sum(x)+a, 532 - sum(x) + b)
  #store p for later
```

```
  p_vec[i] <- p

  #create vectors to store y's corresponding to x=0 and x=1
  n1 = length(x) - sum(x)
  n2 = sum(x)
  y1 <- rep(0, n1)
  y2 <- rep(0, n2)

  #store y values into corresponding x=0 and x=1 vectors
  idx1 = 1
  idx2 = 1
  idx = 1
  for (e in x) {
    if(e == 0){
      y1[idx1] = glucose.dat$Glucose[idx]
      idx1 = idx1 + 1
  }
    else{
      y2[idx2] = glucose.dat$Glucose[idx]
      idx2 = idx2 + 1
    }
    idx = idx+1
  }
  #sample posterior values
  theta1[i] <- rnorm(1, (sum(y1)/sigma1[i-1] +
                           mu/tao)/(n1/sigma1[i-1] + 1/tao),
                     sqrt(1/(n1/sigma1[i-1] + 1/tao)))
  theta2[i] <- rnorm(1, (sum(y2)/sigma2[i-1] +
                           mu/tao)/(n2/sigma2[i-1] + 1/tao),
                     sqrt(1/(n/sigma2[i-1] + 1/tao)))
  sigma1[i] <- rinvgamma(1, (n1+v)/2, (v*gamma
                    +sum((y1-theta1[i])^2))/2)
  sigma2[i] <- rinvgamma(1, (n2+v)/2, (v*gamma
                    +sum((y2-theta2[i])^2))/2)
}
mean(p_vec)
```

```
## [1] 0.04424512
```

```
mean(theta1)
```

```
## [1] 121.0113
```

```
mean(sigma1)
```

```
## [1] 964.4911
```

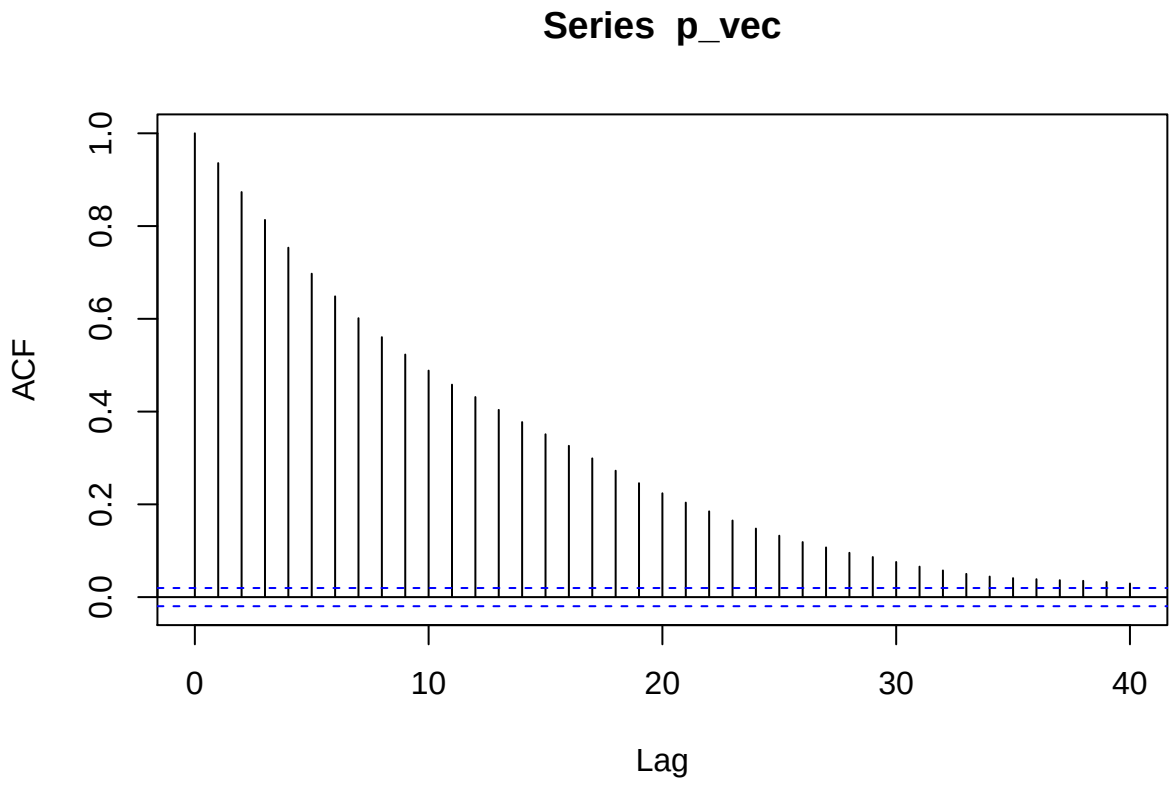
```
mean(theta2)
```

```
## [1] 120.6474
```

```
mean(sigma2)
```

```
## [1] 1122.439
```

```
acf(p_vec)
```

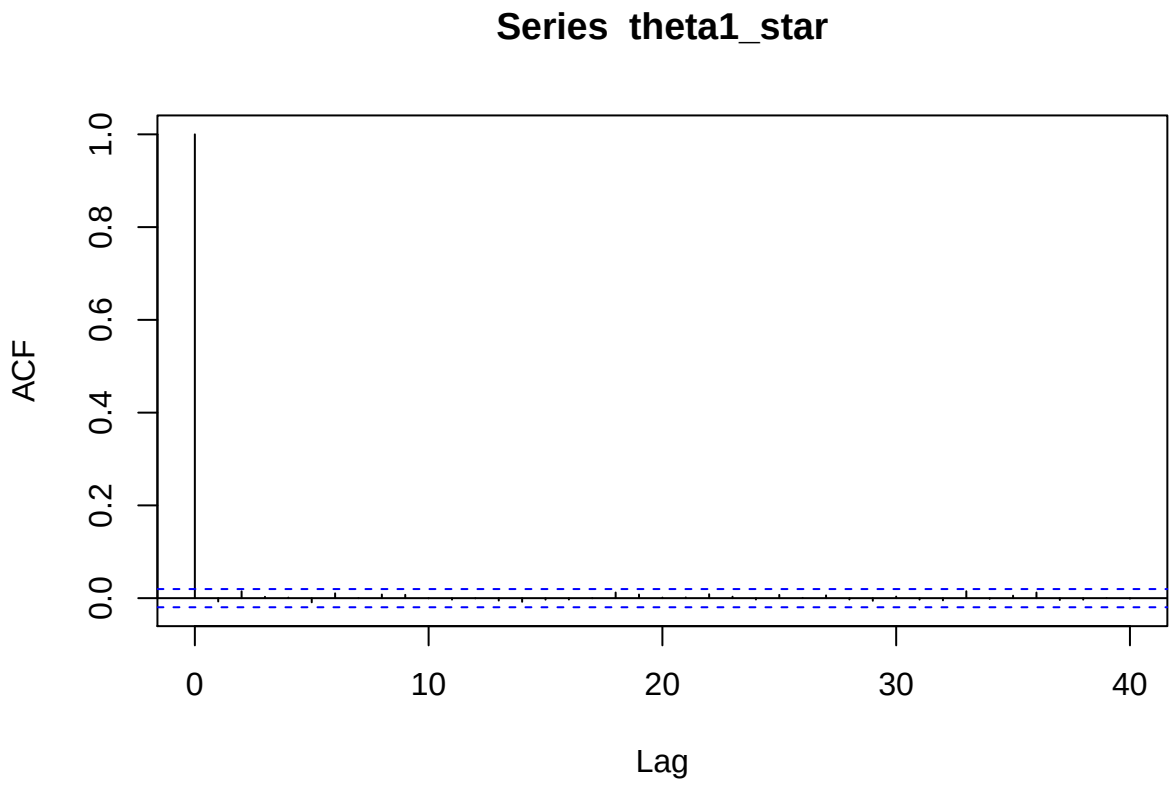



```
effectiveSize(p_vec)
```

```
##     var1
## 384.0812
```

```
theta1_star <- rep(0, 10000)
theta2_star <- rep(0, 10000)
#assign values to theta_star vecs
for (i in 1:10000) {
  theta1_star[i] <- min(theta1[i], theta2[i])
  theta2_star[i] <- max(theta1[i], theta2[1])
}
```

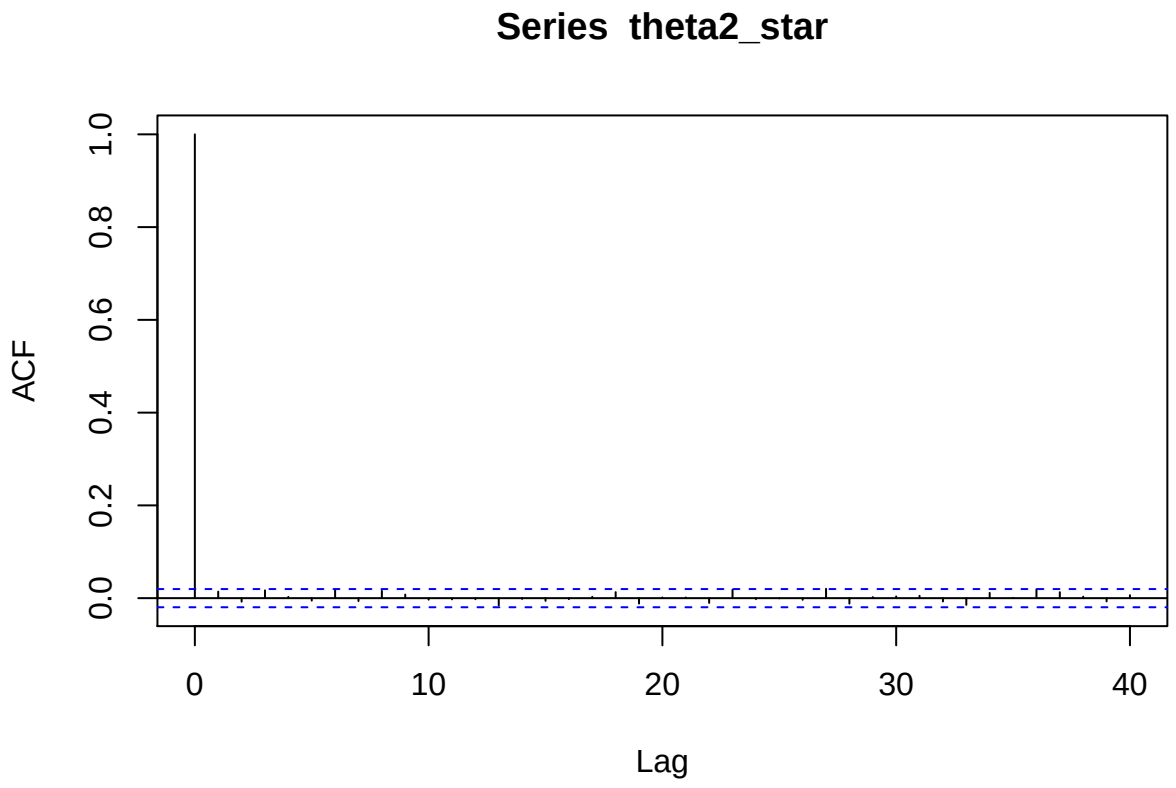
```
#Knitted pdf and rmd show very different outputs
acf(theta1_star)
```




```
effectiveSize(theta1_star)
```

```
##  var1
## 10000
```

```
acf(theta2_star)
```



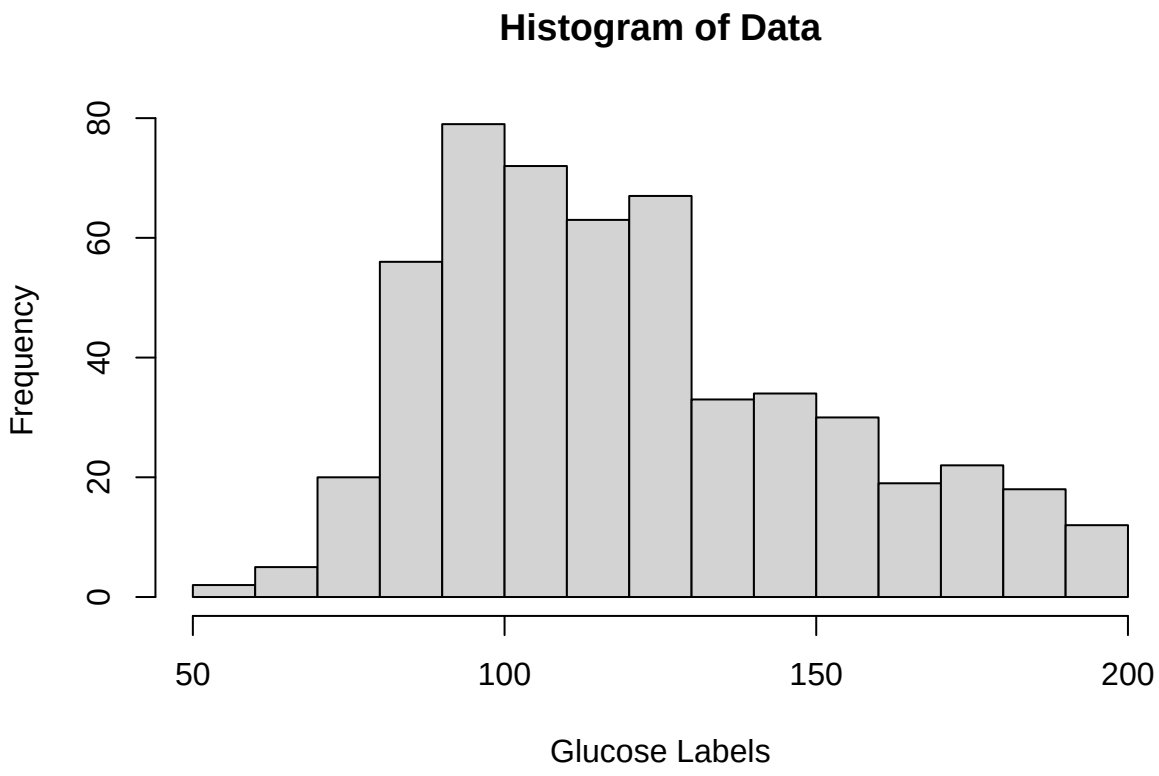

```
effectiveSize(theta2_star)
```

```
##  var1
## 10000
```

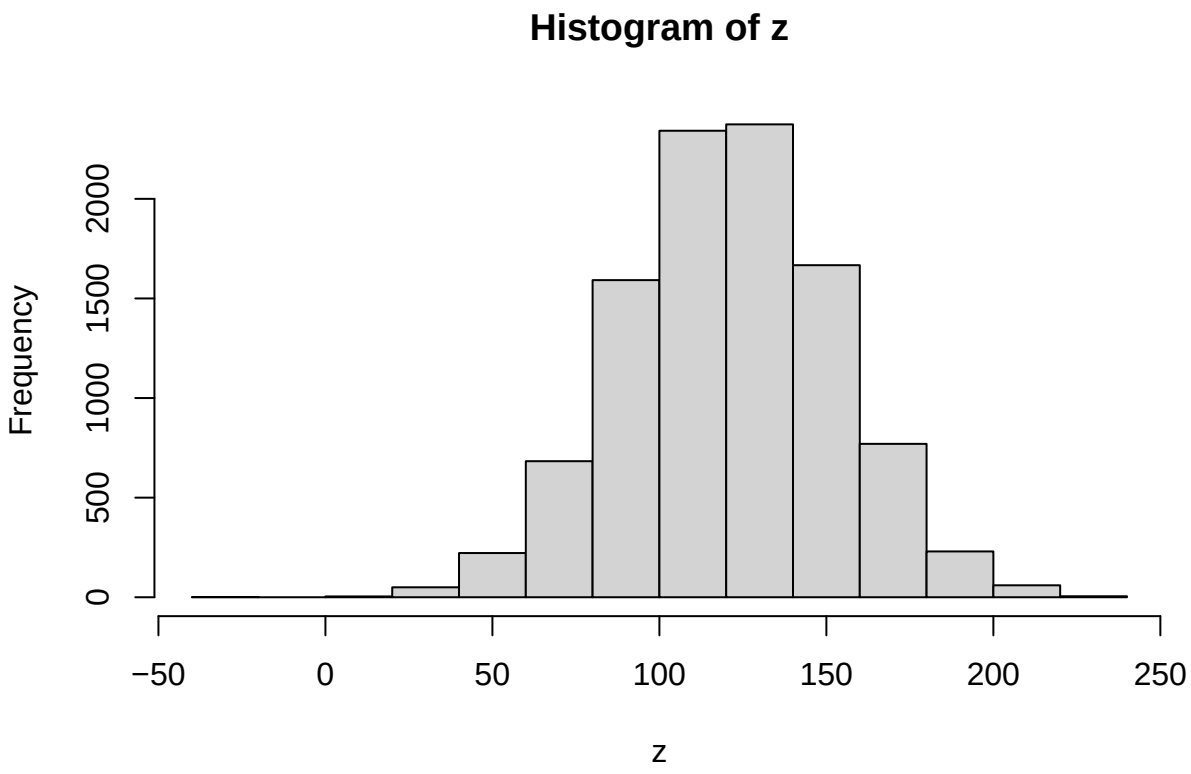
Part D

```
x <- rep(0, 10000)

for (i in 2:10000) {
  x[i] <- rbernoulli(1, p_vec[i])
}

z <- rep(0, 10000)
for (i in 1:10000) {
  if(x[i] == 0){
    z[i] <- rnorm(1, theta1[i], sqrt(sigma1[i]))
  }
  else{
    z[i] <- rnorm(1, theta2[i], sqrt(sigma1[i]))
  }
}
```

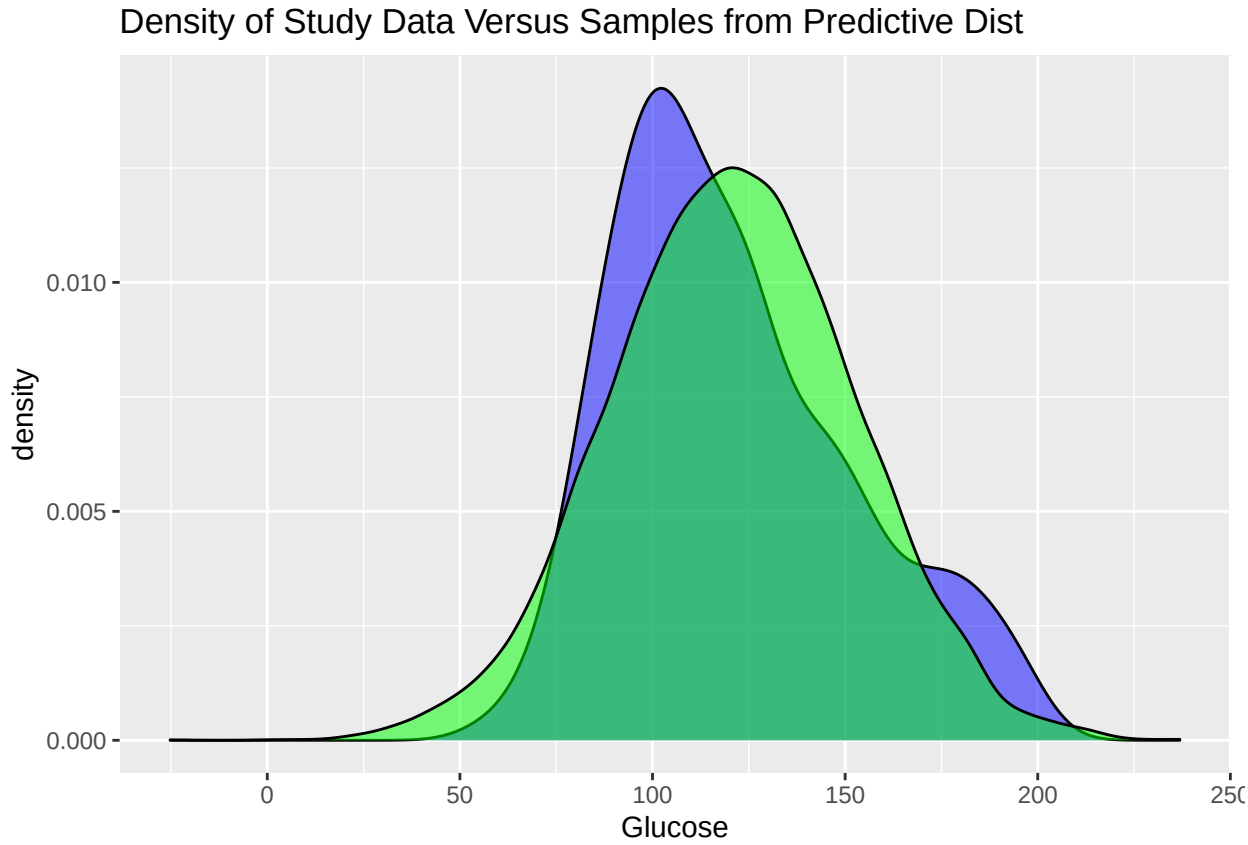
```
hist(glucose.dat$Glucose, main = "Histogram of Data", xlab = "Glucose Labels")
```




```
hist(z)
```




```
ggplot(data = glucose.dat, aes(x = Glucose))+
  geom_density(fill = 'blue', alpha = .5)+
  geom_density(data = data.frame(z), aes(x = z), fill = 'green', alpha = .5)+
  labs(title = "Density of Study Data Versus Samples from Predictive Dist")
```



The mixture model does not appear to be a good fit for the glucose data. The glucose data appears unimodal and skewed right. There is a large swath of glucose data on the right that is not covered by the predictive distribution. The left half of the glucose data has a higher mode and appears to be more centered around 100 whereas the mixture model is a mixture of two normal distributions more centered around 120 and 121. The mixture model is also slightly bimodal which means the two distributions are most likely fundamentally different.