

Q1

The likelihood function is:

And the prior distribution is:

So the posterior distribution is:

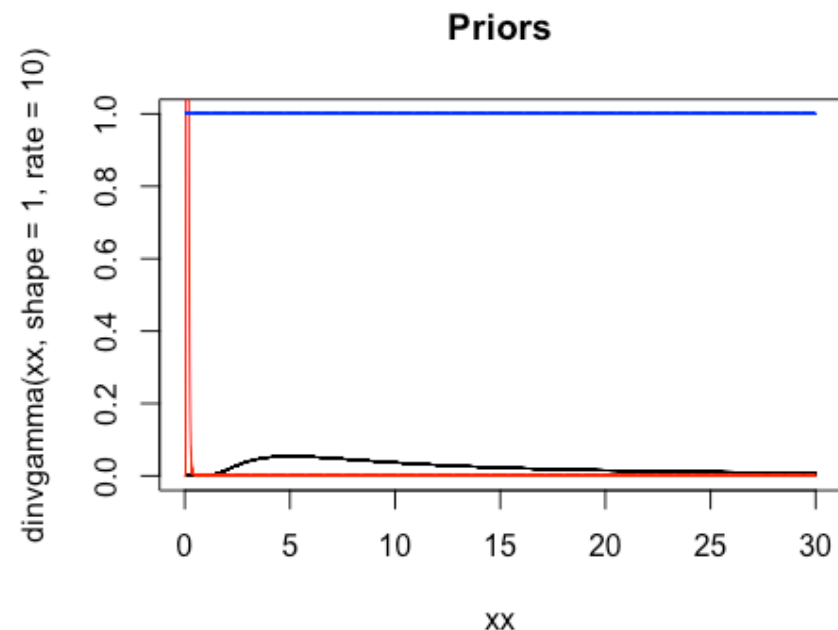
So the posterior density is:

```
library(invgamma)
set.seed(440)
data = rweibull(24, shape=1/2, scale=64)
n = length(data)
p = matrix(0, nrow = 3, ncol = 2)
p[1,] = c(1,10)
p[2,] = c(10,1)
p[3,] = c(NA,NA)
m = p+cbind(rep(n,3), rep(sum(data^0.5),3))
m[3,1] = n-1
m[3,2] = sum(data^0.5)
alpha = 0.05
level = c(alpha/2, 1-alpha/2)
rr = rbind(qinvgamma(level,shape = m[1,1],rate = m[1,2]),
           qinvgamma(level,shape = m[2,1],rate = m[2,2]),
           qinvgamma(level,shape = m[3,1],rate = m[3,2]))

result=as.matrix(cbind(p,m,m[,2]/(m[,1]-1),rr))
dimnames(result) = list(c("1", "2", "3"), c("Prior1", "Prior2", "Post1", "Post2", "Mean", "Lower",
"Upper"))

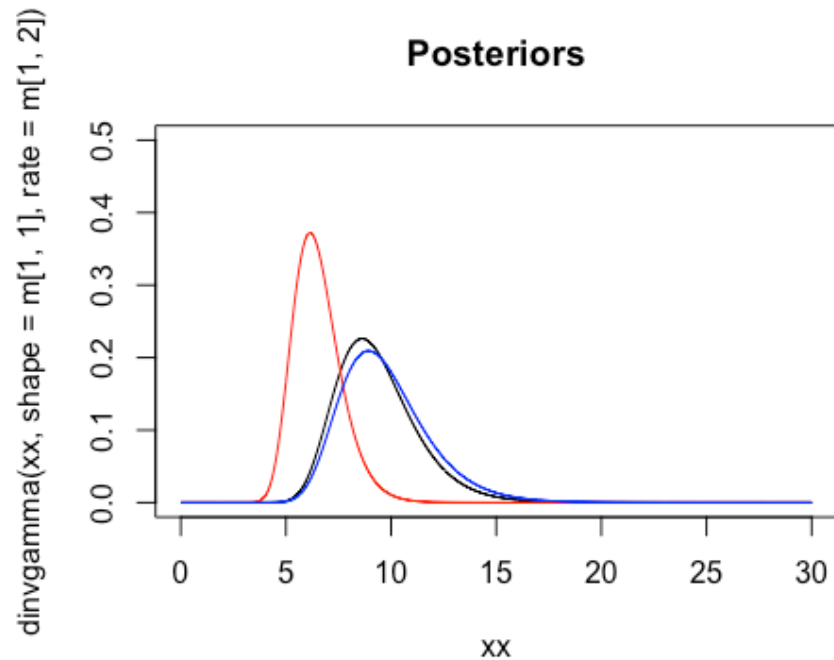
result
## Prior1 Prior2 Post1 Post2 Mean Lower Upper
## 1 1 10 25 224.2029 9.341786 6.278417 13.857920
## 2 10 1 34 215.2029 6.521299 4.643570 9.139683
## 3 NA NA 23 214.2029 9.736494 6.430923 14.691528
xx=seq(0,30,0.001)

plot(xx, dinvgamma(xx, shape = 1, rate = 10), type="l", main = "Priors",ylim = c(0,1) )
lines(xx, dinvgamma(xx, shape = 10, rate = 1), type = "l",col = 'red')
lines(xx, rep(1,length(xx)), type = "l", col = 'blue')
```



```
xx=seq(0,30,0.01)
plot(xx, dinvgamma(xx, shape=m[1,1], rate = m[1,2]), type = "l", main = "Posteriors",ylim =
```

```
c(0,0.5))
lines(xx, dinvgamma(xx, shape=m[2,1], rate = m[2,2]), type = "l",col = 'red')
lines(xx, dinvgamma(xx, shape=m[3,1], rate = m[3,2]), type = "l",col = 'blue')
```



```
ga_est = 1/n*sum(data^0.5)
```

```
loglik<-function(gamma = NULL, datax = NULL){
  val = numeric(length(gamma))
  n = length(data)
  for (i in 1:length(gamma)){
    val[i] = n*log(0.5) + sum(log(datax)) -n*log(gamma)- sum(datax^0.5)/gamma
  }
  return (val)
}
```

```
loglik.ratio <- function(x = NULL, datax = NULL, gamma.est = NULL, q0 = NULL){
  ratio = -2 * (loglik(gamma = x, datax = datax) - loglik(gamma = gamma.est, datax = datax))
  val = ratio - q0
  return(val)
}
```

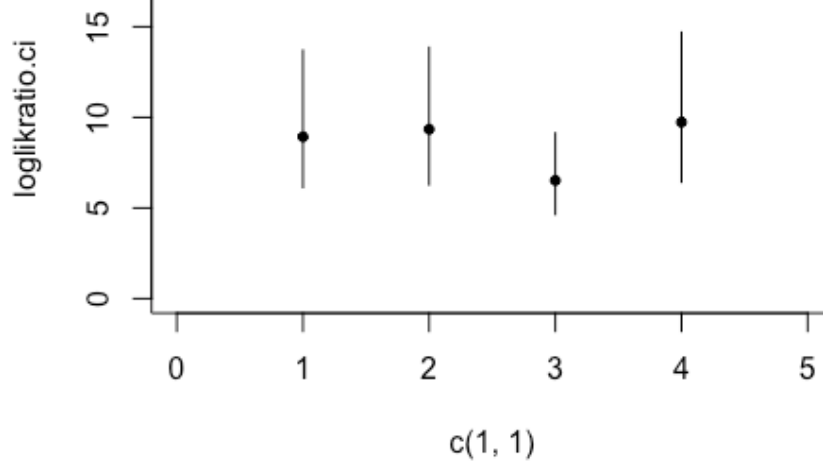
```
temp1 = uniroot(f=loglik.ratio, interval = c(0.1, ga_est), datax = data, gamma.est = ga_est,
q0=3.841459)
```

```
temp2 = uniroot(f = loglik.ratio, interval = c(ga_est,30), datax = data, gamma.est = ga_est, q0 =
3.841459)
```

```
loglikratio.ci = c(temp1$root, temp2$root)
loglikratio.ci
## [1] 6.133611 13.701428
plot(x = c(1,1), y = loglikratio.ci,xlim = c(0,5),ylim = c(0,20),type = 'l')
```

```
for (i in 1:3){
  lines(x = c(i,i)+1, y = rr[i,]) }
```

```
points(1:4, c(ga_est, m[,2]/(m[,1]-1)), pch = 20)
```



```
r2 = rbind(loglikratio.ci, rr)
```

```
r2 = as.matrix(cbind(c(ga_est, m[,2]/(m[,1]-1)),r2))
```

```
dimnames(r2) = list(c('loglikelihood ratio', '1', '2', '3'))
```

```
r2
##           [,1]  [,2]  [,3]
## loglikelihood ratio 8.925120 6.133611 13.701428
## 1           9.341786 6.278417 13.857920
## 2           6.521299 4.643570  9.139683
## 3           9.736494 6.430923 14.691528
```

As can be seen the confidence intervals given by likelihood ratio method, model1(,), and model3() are similar, which are wider than that given by model2(,)

The prior is

So the density is given by:

```
prob <- function(x) {
  if (x<0) return (0)

  else return (dinvgamma(x, shape = 23, rate = 225))
}

mcmc <- function(n = NULL, sigma = 1, x0){
  x = numeric(n)

  x[1] = x0

  for (i in 2:n){
    y = rnorm(1, x[i-1], sd = sigma)
    u = runif(1)
    accept = prob(y) / prob(x[i-1])

    if (u < accept) {x[i] = y}

    else {x[i] = x[i-1]}
  }

  return (x)
}

set.seed(440)

n = 10^4

x0 = 8.9

sigma = c(0.1, 1, 5, 20)

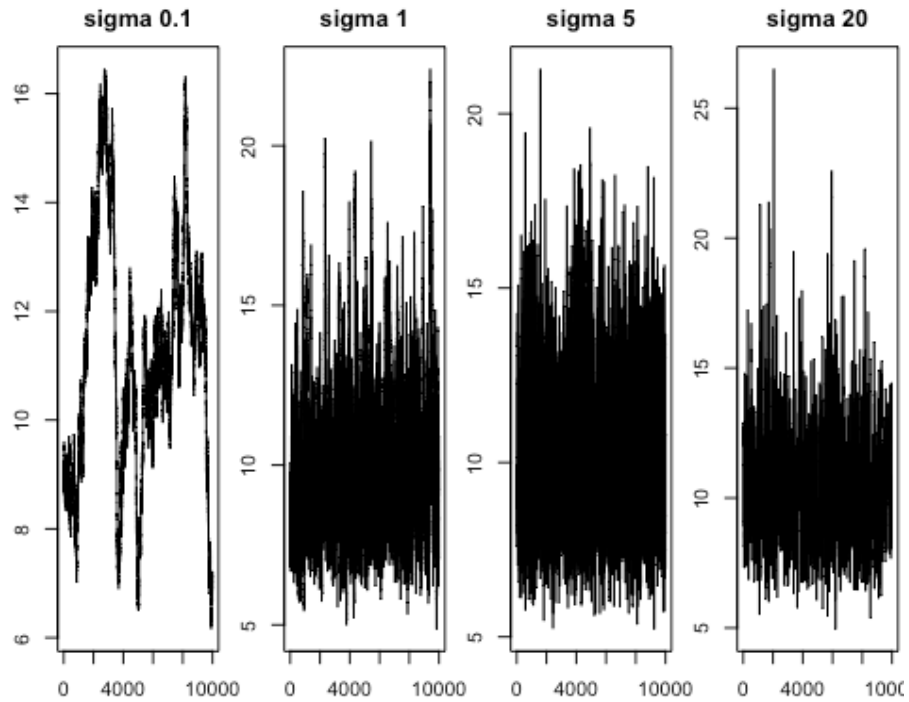
m2 = matrix (0, nrow = n, ncol = length(sigma))

for (i in 1:length(sigma)) {
```

```
m2[,i] = mcmc(n = n, sigma = sigma[i], x0=x0)
}
```

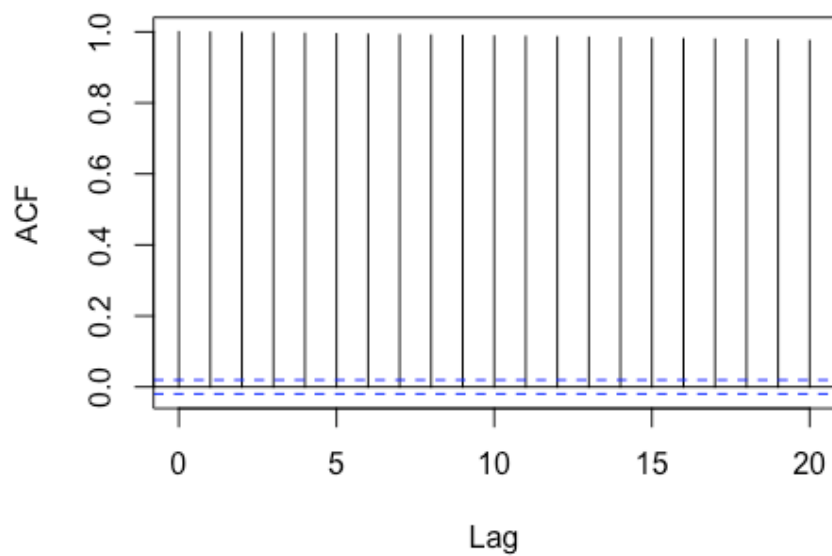
```
par(mfrow = c(1,length(sigma)), mar = 2.5 * c(1,1,1,0.1))
```

```
for (i in 1:length(sigma)) plot(m2[,i], main = paste('sigma', sigma[i]), type='l')
```

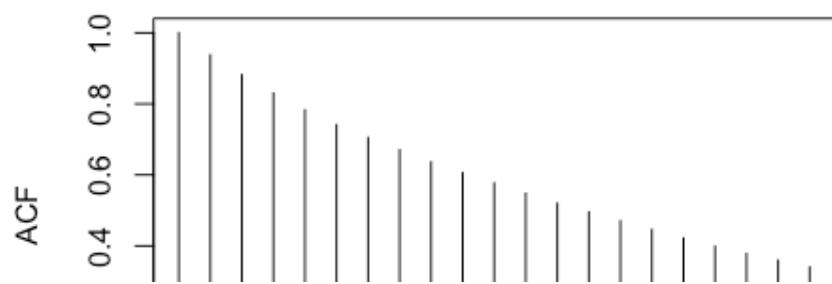


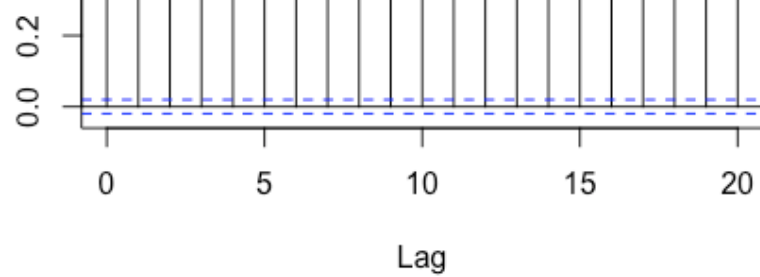
```
for (i in 1:length(sigma)) acf (m2[,i], main = paste('mcmc_sigma',sigma[i]) , lag.max = 20)
```

mcmc_sigma 0.1

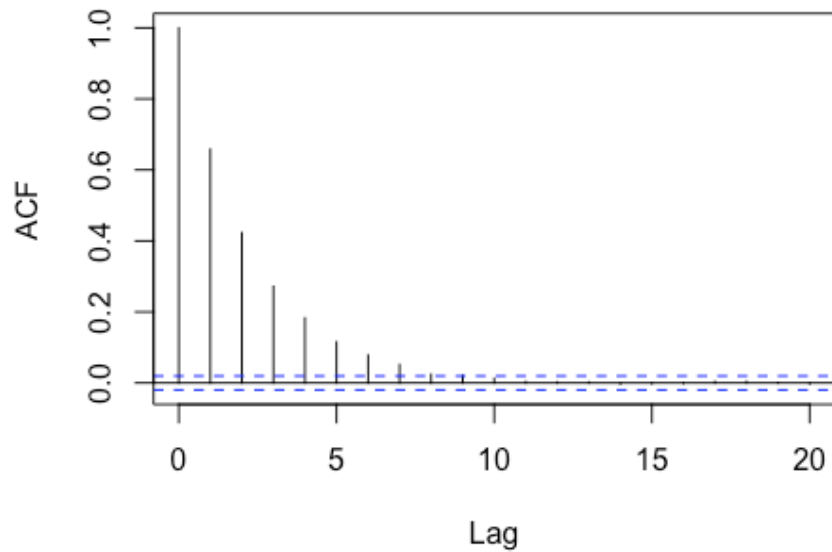


mcmc_sigma 1

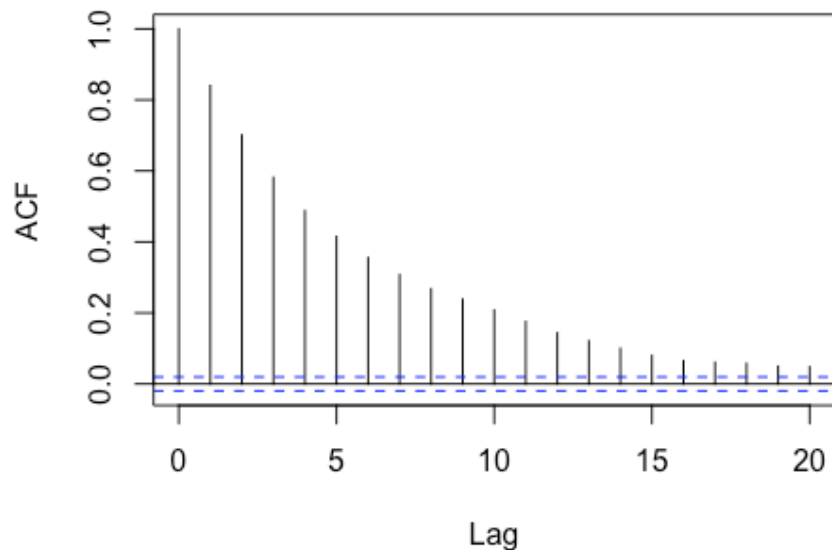




mcmc_sigma 5



mcmc_sigma 20



```
a = rbind(sigma,
  accept = apply(m2, 2, function(x){mean(x[-1] != x[-length(x)] )}),
  mean = apply(m2, 2, mean),
  mixing = apply(m2, 2, function(x){mean(diff(x)^2)}),
  apply(m2, 2, quantile, probs = c(0.025, 0.975)), 4)
```

As can be seen the model with gives the best result. When sigma is smaller than 5, the plots shows large correlation, which indicates a larger number would be appropriate. However, the model with has higher correlation compared with that with . The true value of CI is (6.1336, 13.7014). So, model with and model with give the best estimate CI.

