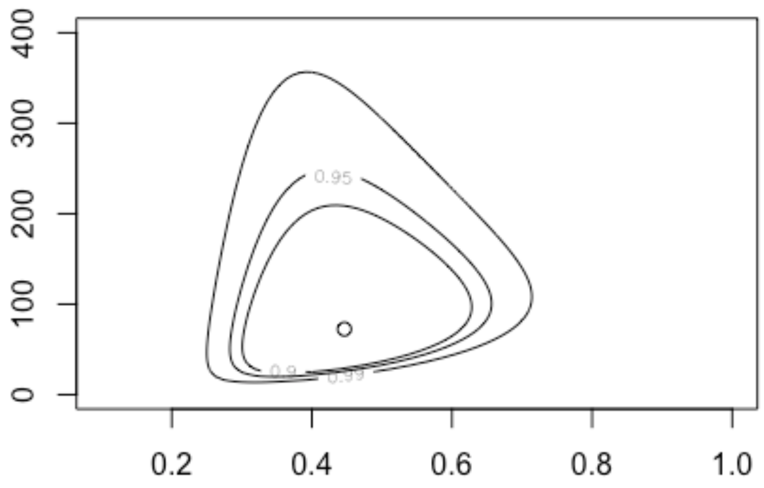
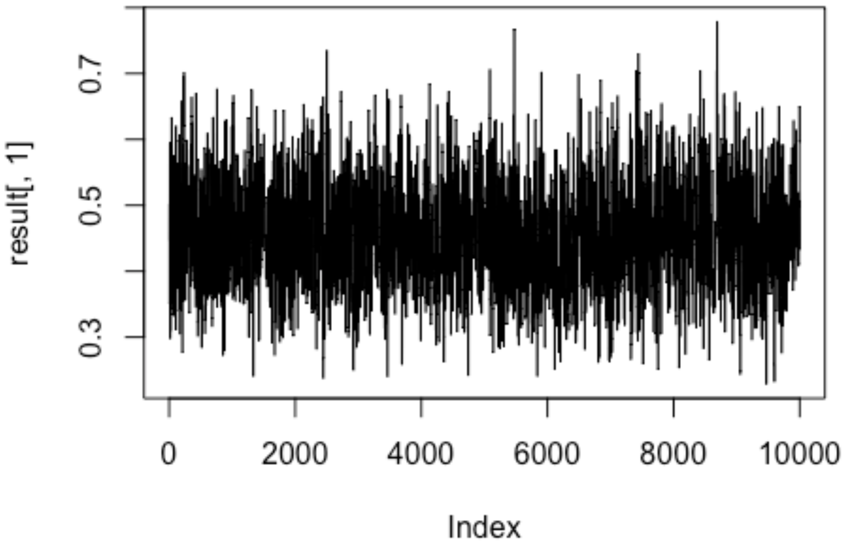
q3

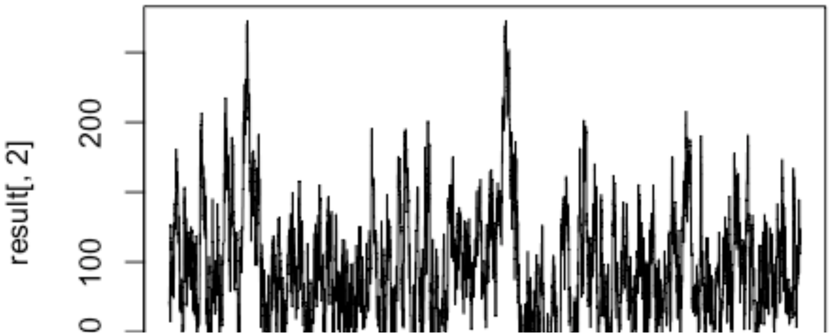
```
library('invgamma')
set.seed(440)
x = rweibull(24,0.5,64)
loglik <- function(shape = NULL, scale = NULL, x = NULL){
 val = matrix(0, nrow = length(shape), ncol = length(scale))
 for (i in 1:length(shape)){
  for (j in 1:length(scale)){
   val[i,j] = sum(dweibull(x, shape = shape[i],scale = scale[j], log = TRUE))
  }
 }

 return (val)
}

loglik2 <- function(p = NULL, datax = NULL){
 val = sum(dweibull(datax, shape = p[1], scale = p[2], log = TRUE))

 return (val)}

temp = optim(par = c(1,1), loglik2, datax = x, control = list(fnscale = -1))
## Warning in dweibull(datax, shape = p[1], scale = p[2], log = TRUE): NaNs
## produced

## Warning in dweibull(datax, shape = p[1], scale = p[2], log = TRUE): NaNs
## produced
mle = temp$value
mle_shape = temp$par[1]
mle_scale = temp$par[2]
s1 = seq(0.1,1,length.out = 100)
s2 = seq(0.1,400,length.out = 100)
r = loglik(shape = s1, scale = s2, x = x)

contour(x = s1, y = s2, z = -2*(r-mle), levels = qchisq(c(0.9,0.95,0.99), df = 2), labels =
c('0.9','0.95','0.99' ))

points(x = mle_shape, y = mle_scale)
```

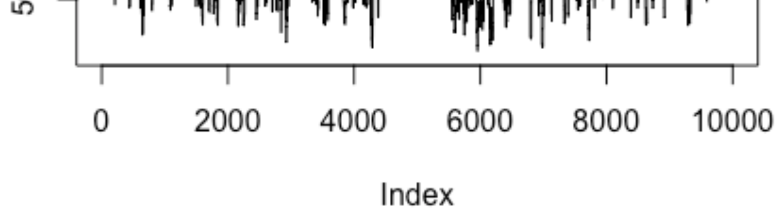


```
posterior <- function(shape = NULL, scale = NULL, datax = NULL ){
 if (shape <= 0 | scale <= 0){r = 0}
 else {l = length(datax)
 r = shape^l/scale^l*prod(datax^(shape-1))/scale^(l*(shape-1))*exp(-
 r = shape^n/scale^n*prod(datax^(shape-1))/scale^(n*(shape-1))*exp(-
sum(datax^shape)/scale^shape)}

 return(r)
}

random_walk <- function(n = NULL, xx=c(0,0), sigma = c(0,0)){
 m = matrix(0, nrow = n, ncol = 2)
 m[1,] = xx
 for (i in 2:n){
   val = rnorm(2, m[i-1,], sd = sigma)
   val2 = runif(1)
   accept = posterior(shape = val[1], scale = val[2], datax = x)/posterior(shape = m[i-1,1], scale =
m[i-1,2], datax = x)

   if (val2 < accept){m[i,] = val}
   else {
     m[i,] = m[i-1,]
   }
 }
 return (m)
}
set.seed(440)
n = 10^4
result = random_walk(n = n, xx = c(0.446,72.439),sigma = c(0.2,15))
mean(result[-1,1] != result[-n,1])
## [1] 0.3862386
mean(result[-1,2] != result[-n,2])
## [1] 0.3862386
plot(result[,1], type = 'l')
```

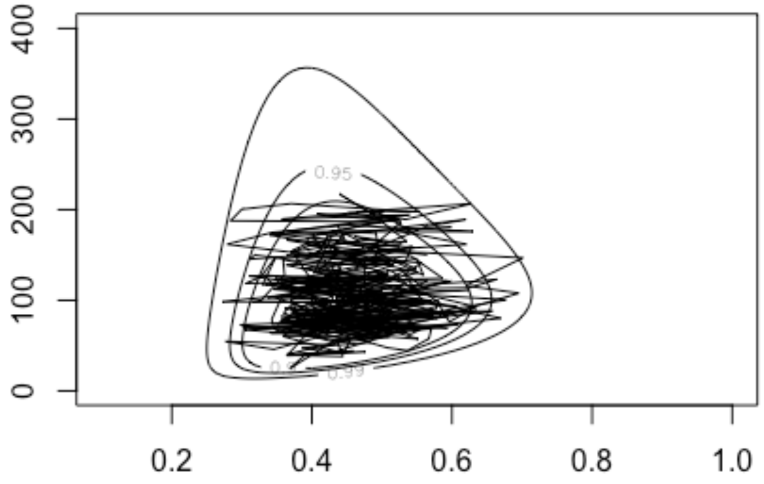


```
plot(result[,2], type = 'l')
```





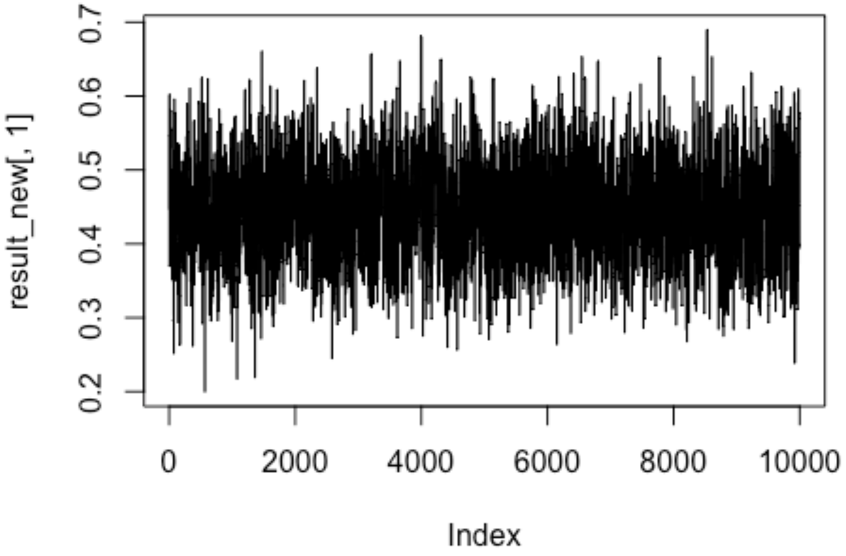
```
contour(x = s1, y = s2, z = -2*(r-mle), levels = qchisq(c(0.9, 0.95, 0.99), df = 2), labels = c('0.9',
'0.95', '0.99'))

lines(result[1:1000,],type = 'l')
```



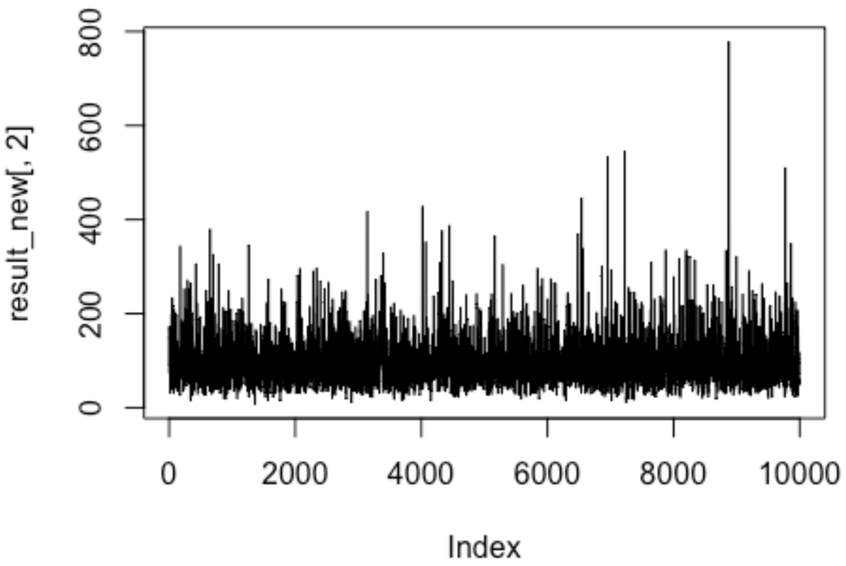
Since $p(, |x) inv(n-1, x_i^) $ we have $ p(|, x) = $ which follows $ inv(n-1,x_i^) $ w.r.t $=
^$

```
alpha_est = 0.44029

prob <- function(alpha = NULL, datax = NULL){
 if (alpha <= 0 )
   return(-Inf)
 else{n = length(datax)
   result = n*log(alpha)+(alpha-1)*sum(log(data))-n*alpha*log(64)-sum(data^alpha)/(64^alpha)

 return (result)}

}
f <- function(n=NULL, xx = c(0,0)){
 m = matrix(0, nrow = n, ncol = 2)
 m[1,] = xx

 for (i in 2:n){
   if (i%%2 == 0){
     val = rnorm(1, mean = alpha_est,sd = sqrt(1/(24/alpha_est^2+sum((x/64)^alpha_est*
(log(x/64))^2))))
     val2 = runif(1)

     accept = posterior(shape = val, scale = m[i-1,2], datax = x)/posterior(shape = m[i-1], scale =
m[i-1,2], datax = x)*dnorm(m[i-1], mean = alpha_est, sd =
sqrt(1/(24/alpha_est^2+sum((x/64)^alpha_est*(log(x/64))^2))))/dnorm(val, mean = alpha_est, sd
= sqrt(1/(24/alpha_est^2+sum((x/64)^alpha_est*(log(x/64))^2))))

     if (val2 < accept) m[i,] = c(val, m[i-1,2])
     else m[i,] = m[i-1,]
   }

   else {
     val_new = (rinvgamma(1, shape=23, rate=sum(x^m[i-1,1])))^(1/m[i-1,1])
```

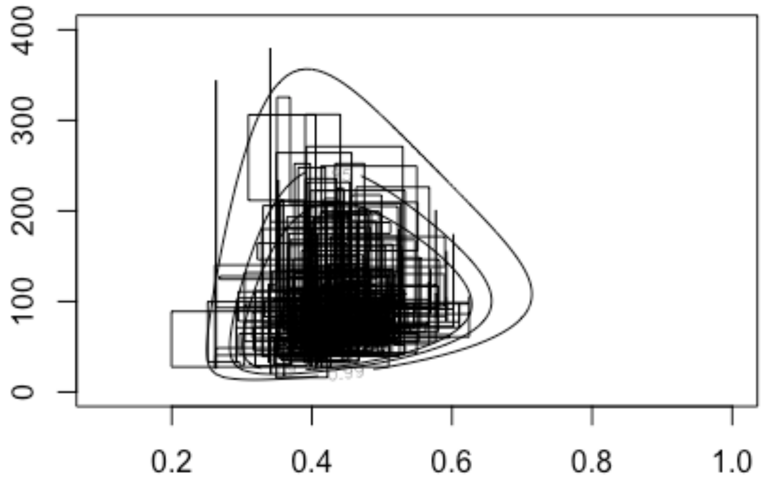
```
    val_new = (rinvgamma(1, shape=25, rate=sum(x^m[i-1,1])))^(1/m[i-1,1])
    val2_new = runif(1)
    accept = 1
    if (val2_new < accept){
      m[i,] = c(m[i-1,1], val_new)}
    else m[i,] = m[i-1,]


  }
 }
 return (m)
}
set.seed(440)
num = 10000
result_new = f(n = num, xx = c(0.446, 72.439))
mean(result_new[-1,1] != result_new[-num, 1])
## [1] 0.4168417
mean(result_new[-1,2] != result_new[-num, 2])
## [1] 0.49995
plot(result_new[,1], type = 'l')
```



```
plot(result_new[,2], type = 'l')
```



```
contour(x = s1, y = s2, z = -2*(r.mle), levels = qchisq(c(0.9,0.95,0.99), df=2), labels = c('0.9'
contour(x = s1, y = s2, z = -2*(l-lmle), levels = qchisq(c(0.9,0.95,0.99), df=2), labels = c('0.9',
'0.95', '0.99'))

lines(result_new[1:1000,], type = 'l')
```


0
100
200
300
400
0.2
0.4
0.6
0.8
1.0
0.99