

IndeptGamma

Paul M

2/1/2021

This example taken from <http://www.mas.ncl.ac.uk/~ndjw1/teaching/sim/metrop/indep.r> We will write a metropolis-hastings independence sampler for a gamma rv based on normal candidates with the same mean and variance.

If you don't have it already, you will need the coda library:

```
#install.packages("coda")
library(coda)
```

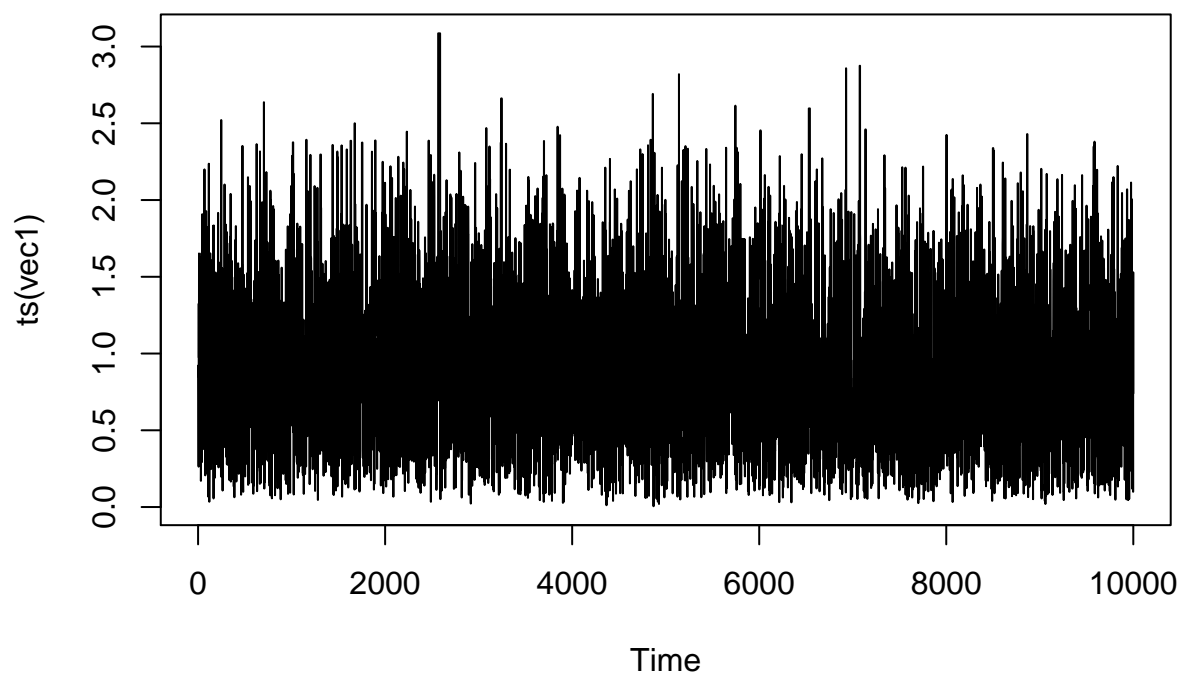
Set the seed and define a function to do this:

```
set.seed(37)
gamm<-function (n, a, b)
{
  mu <- a/b      # the mean of the gamma distribution
  sig <- sqrt(a/(b * b))  # the stadard deviation of the gamma distn
  vec <- vector("numeric", n)  # this is where we are going to put the random variables we generate
  x <- a/b
  vec[1] <- x # We arbitrarily start the MCMC process at the mean
  for (i in 2:n) {
    can <- rnorm(1, mu, sig)
    hprob <- min(1, (dgamma(can, a, b)/dgamma(x,a,b))/(dnorm(can, mu, sig)/dnorm(x, mu, sig))) # wher
    u <- runif(1)
    if (u < hprob)
      x <- can
    vec[i] <- x
  }
  return (vec)
}
```

Here's what happens when we use it:

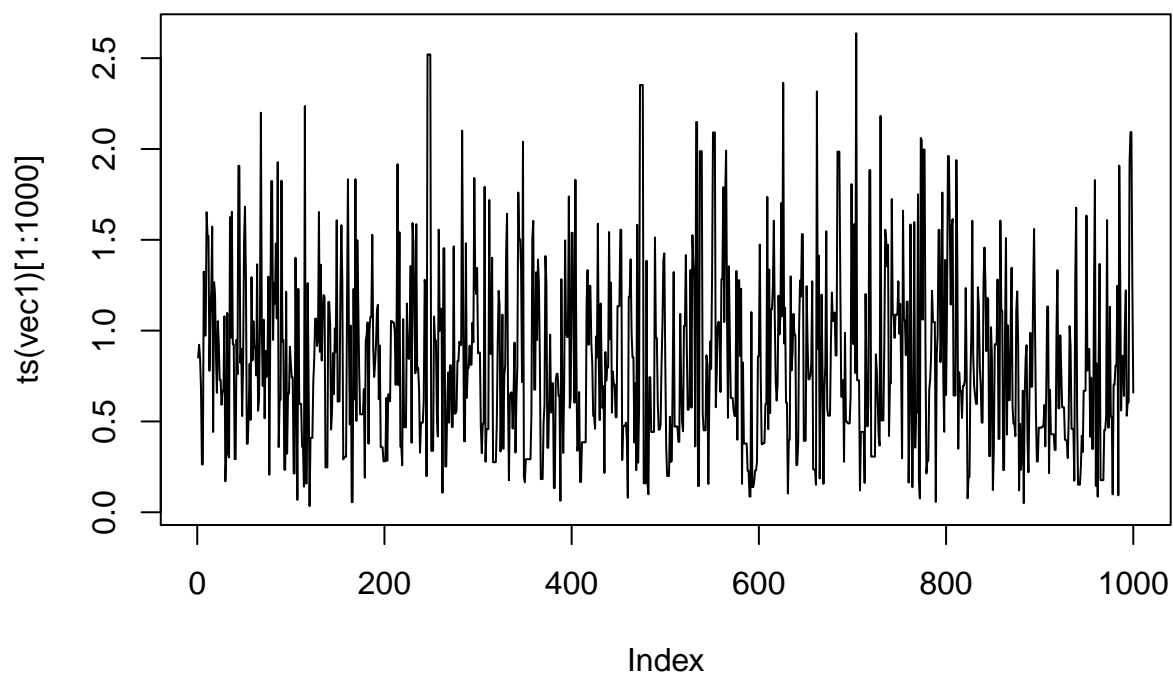
```
vec1<-gamm(10000,2.3,2.7)
vec2<-gamm(10000,2.3,2.7)
plot(ts(vec1),main="vec1")
```

vec1



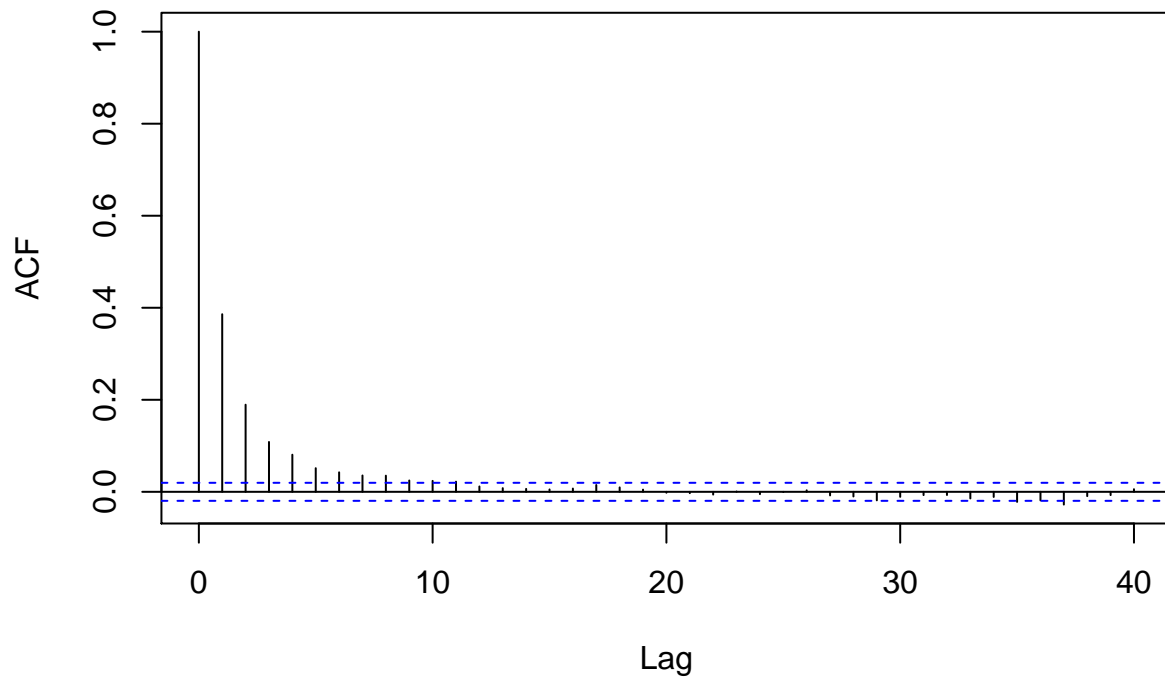
```
plot(ts(vec1)[1:1000],type='l',main="vec1")
```

vec1



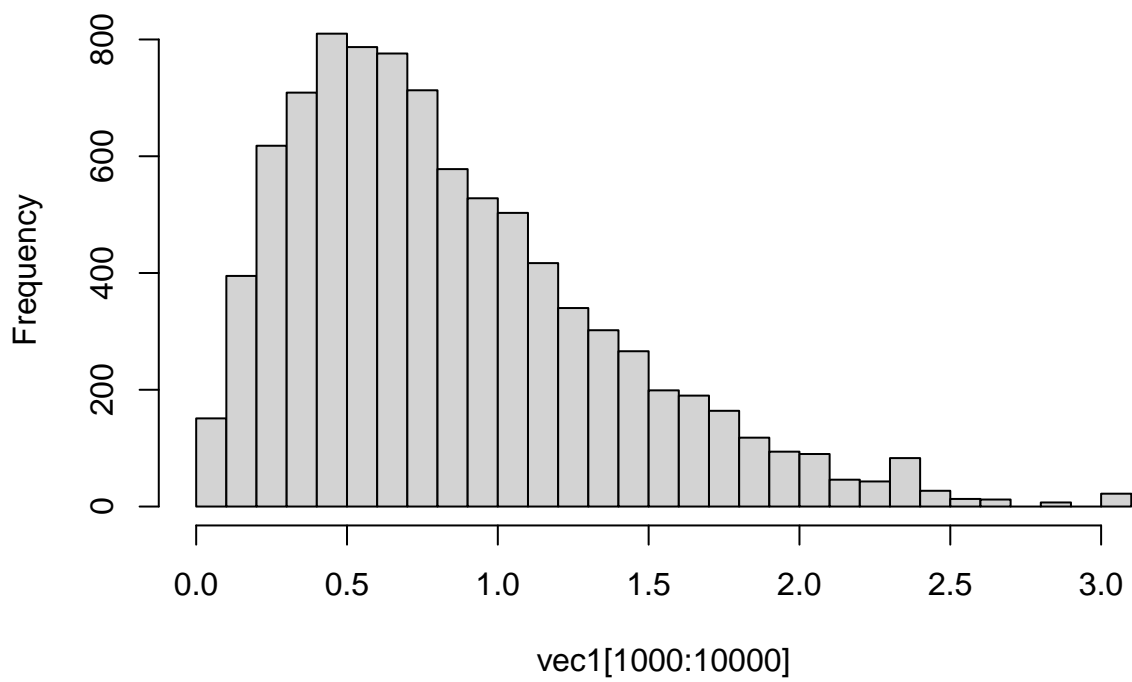
```
acf(vec1)
```

Series vec1

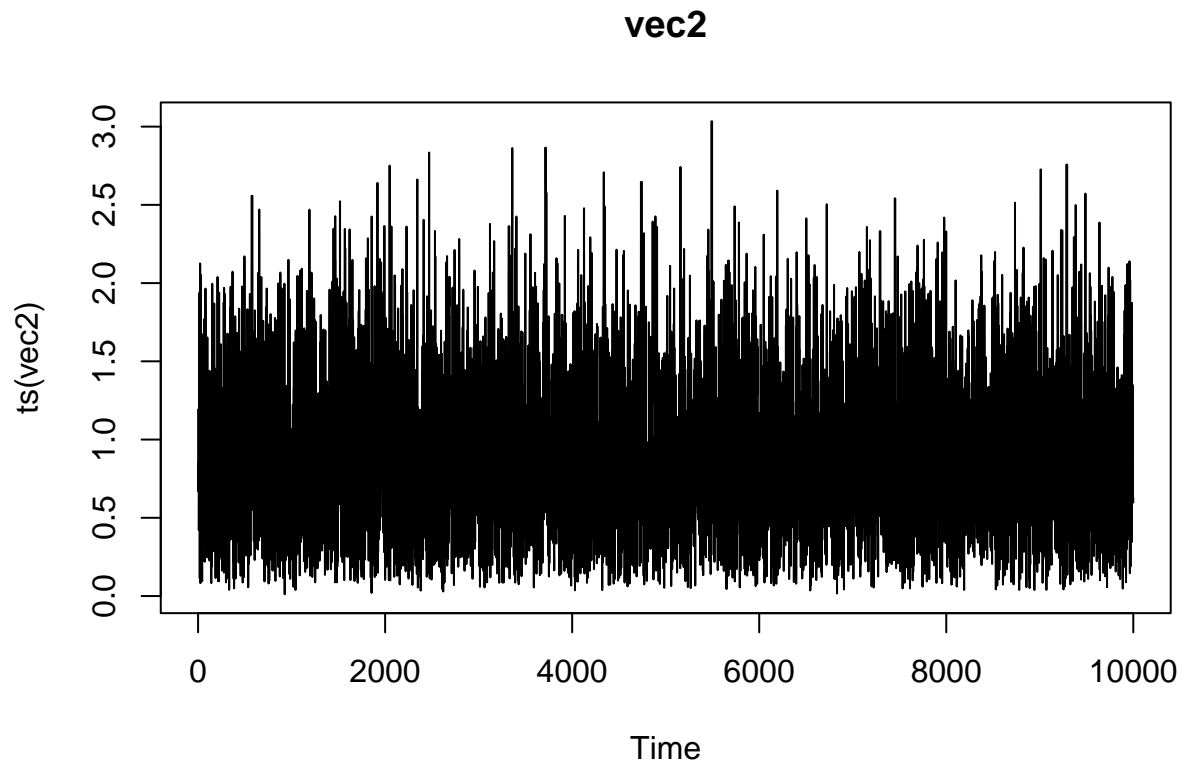


```
hist(vec1[1000:10000],30,main="vec1")
```

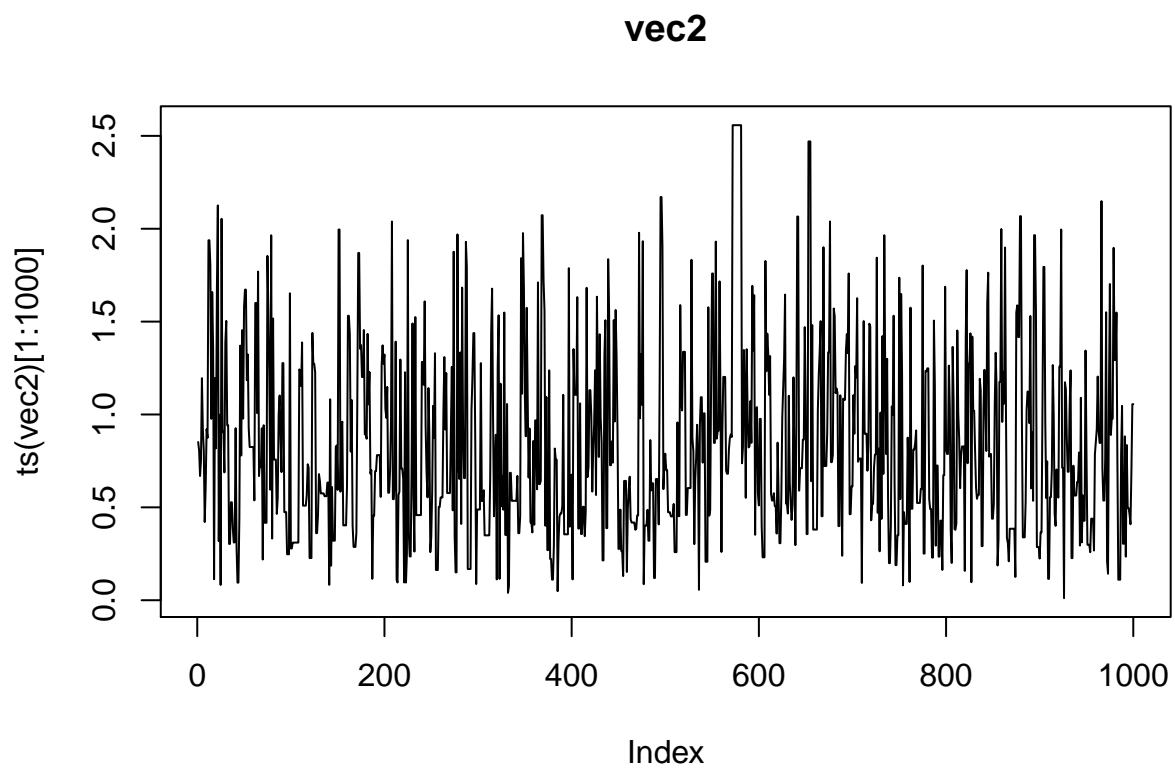
vec1



```
plot(ts(vec2),main="vec2")
```

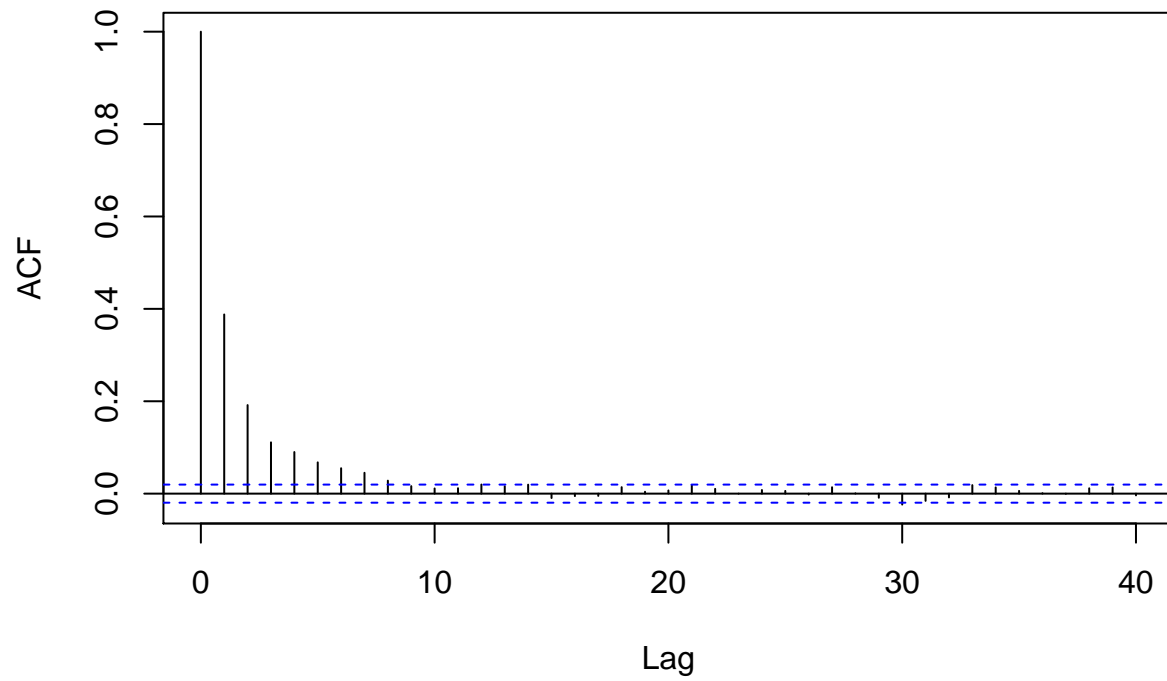


```
plot(ts(vec2)[1:1000],type='l',main="vec2")
```



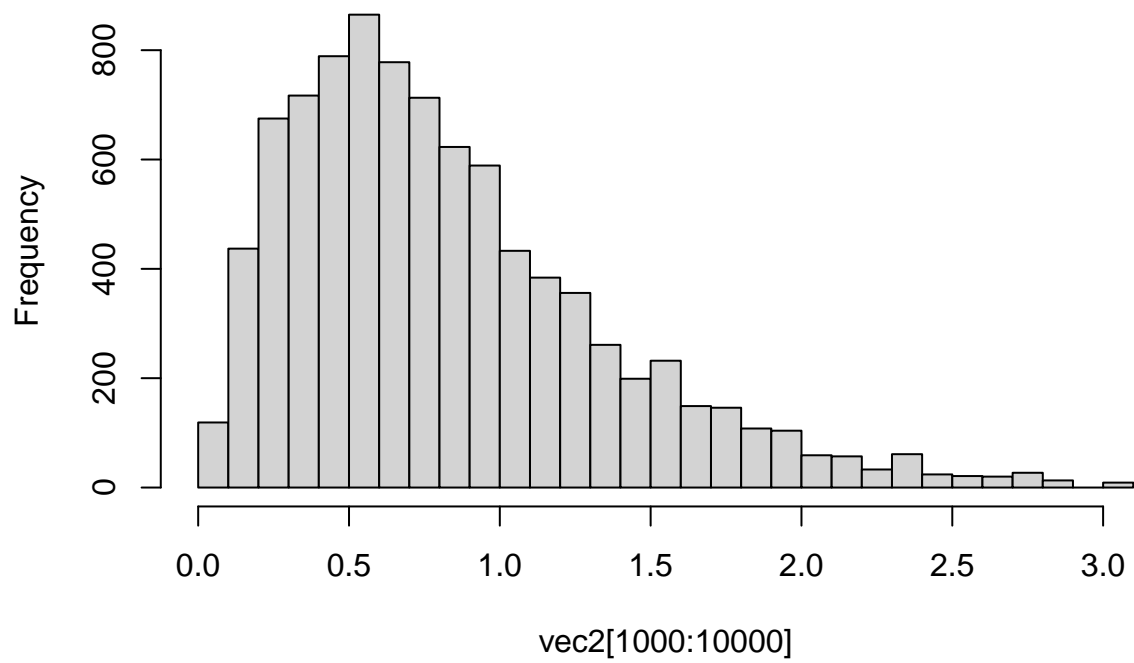
```
acf(vec2)
```

Series vec2

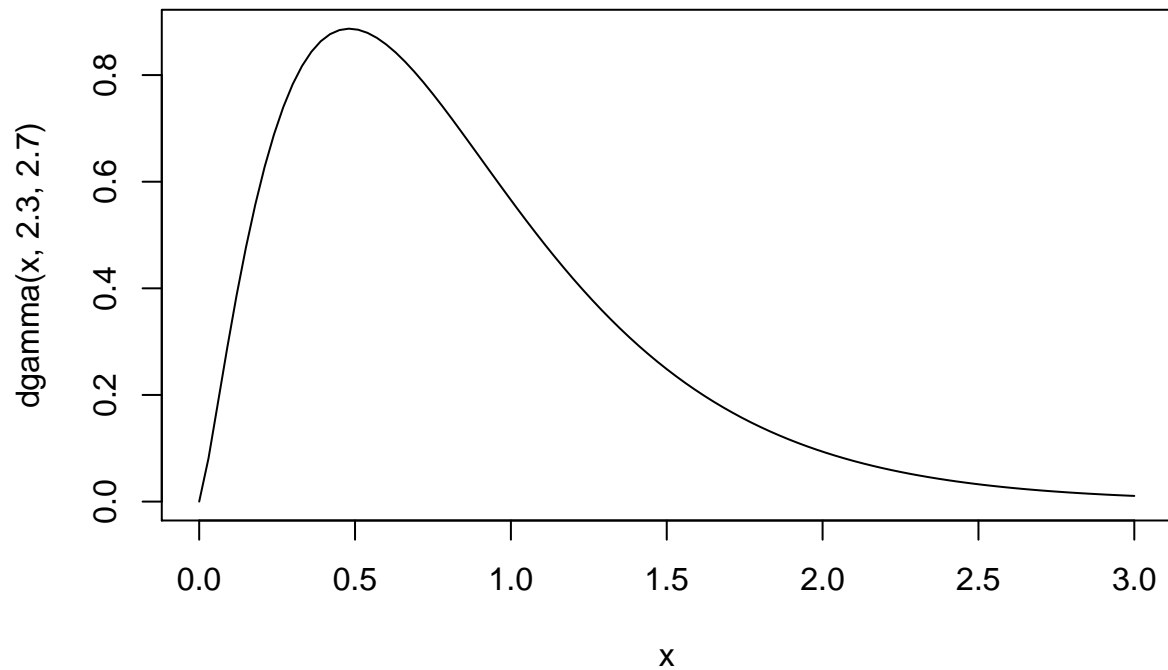


```
hist(vec2[1000:10000],30,main="vec2")
```

vec2



```
curve(dgamma(x,2.3,2.7),from=0,to=3)
```

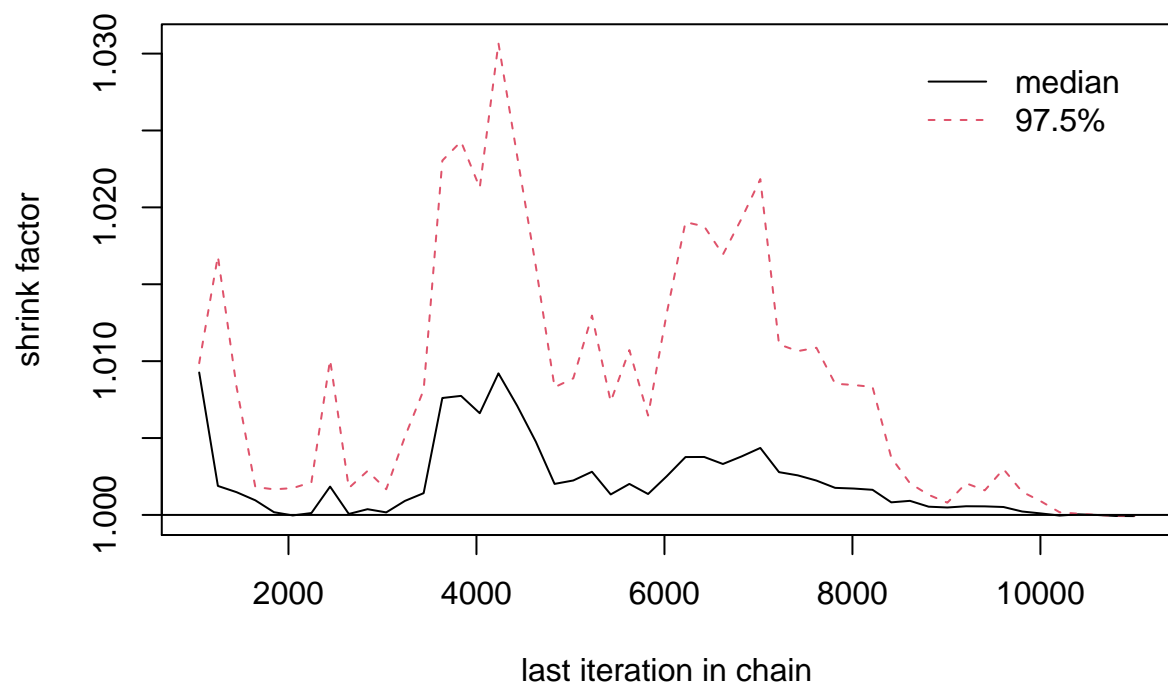


We now convert these runs to MCMC objects and look at Gelman's diagnostics:

```
# convert to mcmc objects, with a burn-in
MCMC1<-mcmc(vec1,start=1000)
MCMC2<-mcmc(vec2,start=1000)

# combine different mcmc chain objects to an mcmc list.
Combined<-mcmc.list(list(MCMC1,MCMC2))

# gelman functions are
gelman.plot(Combined) # for plots
```



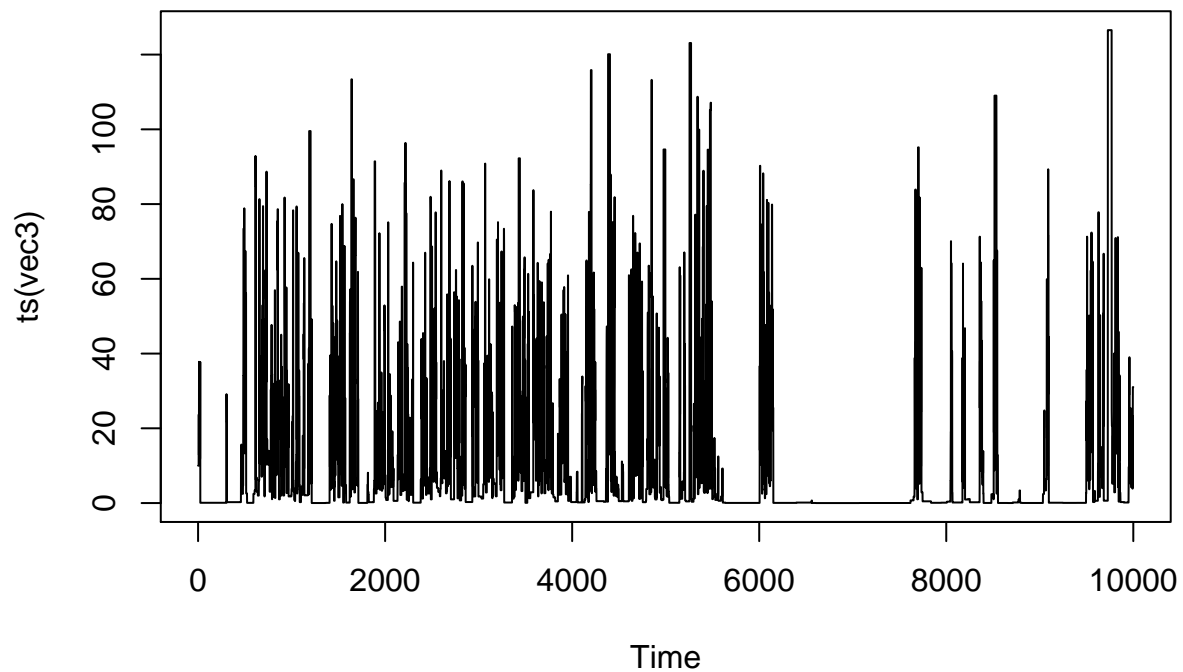
```
print(gelman.diag(Combined)) # for diagnostic values
```

```
## Potential scale reduction factors:  
##  
##      Point est. Upper C.I.  
## [1,]          1          1
```

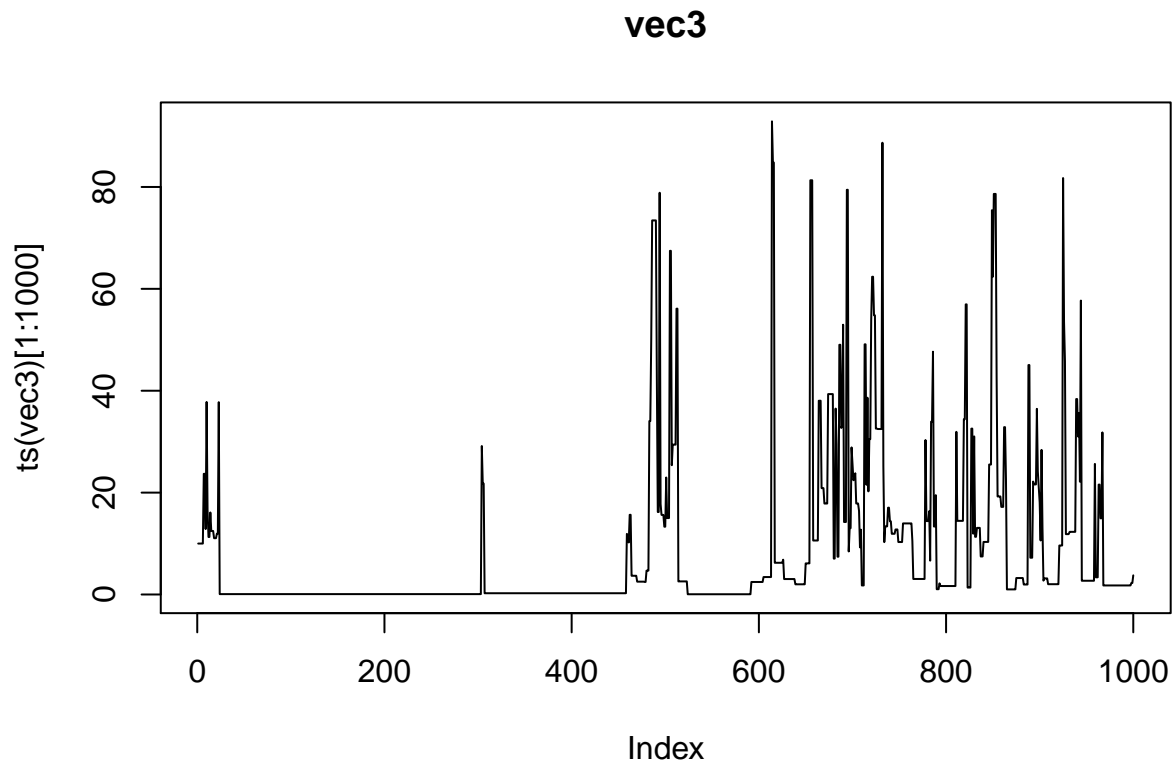
They look decent.

Now let's do the $\text{gamma}(0.1, 0.01)$ case in the same way:

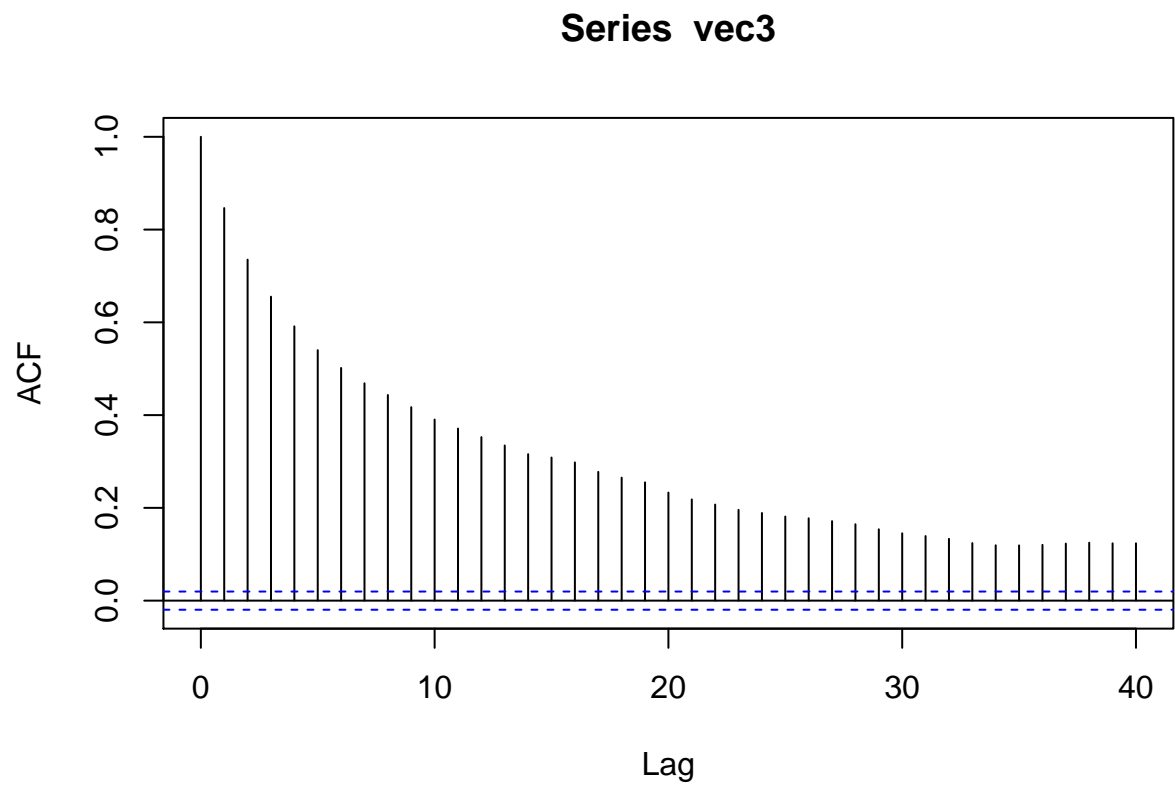
```
vec3<-gamm(10000,0.1,0.01)  
vec4<-gamm(10000,0.1,0.01)  
plot(ts(vec3))
```



```
plot(ts(vec3)[1:1000],type='l',main="vec3")
```

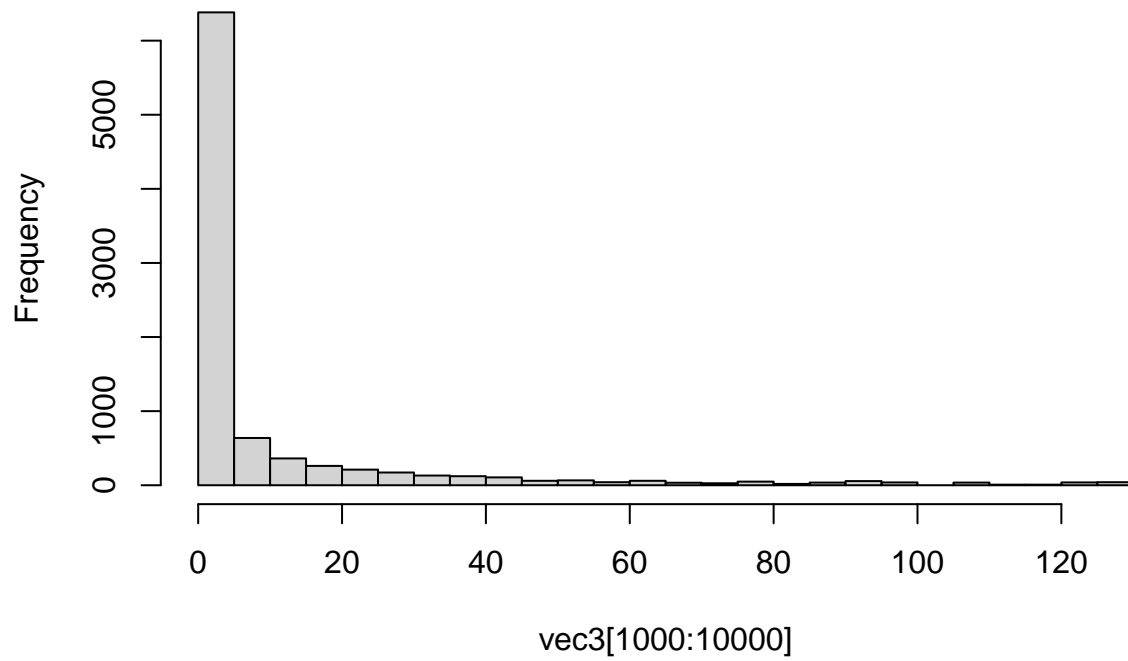


```
acf(vec3)
```



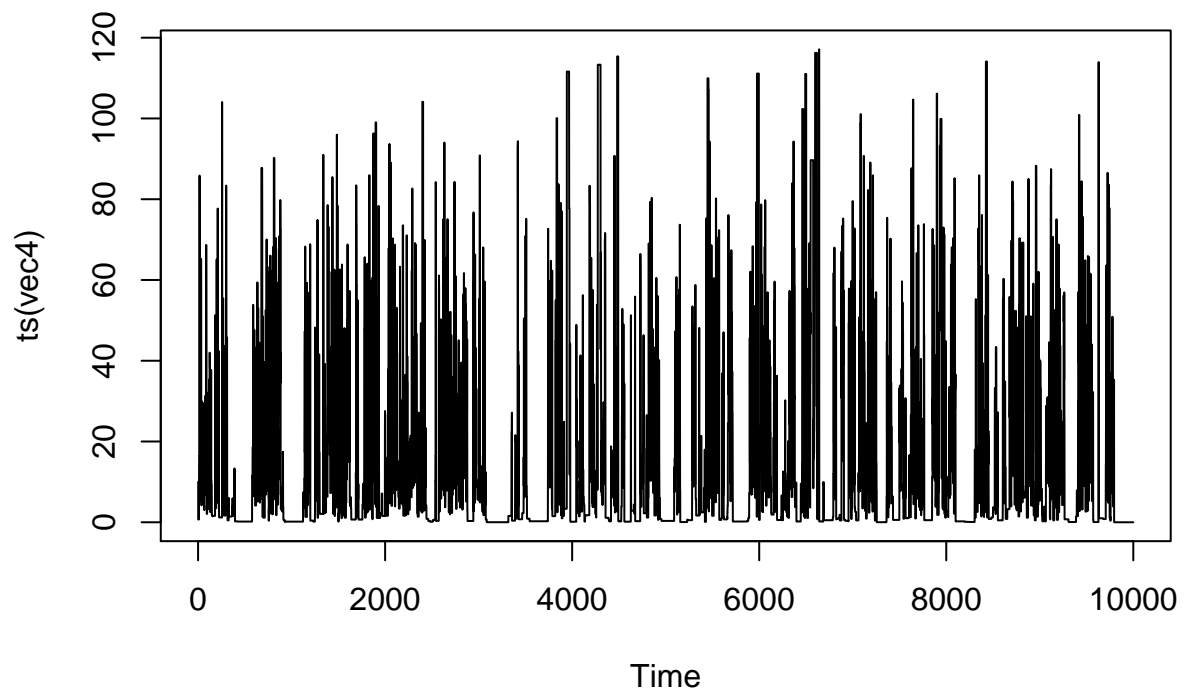
```
hist(vec3[1000:10000],30,main="vec3")
```


vec3



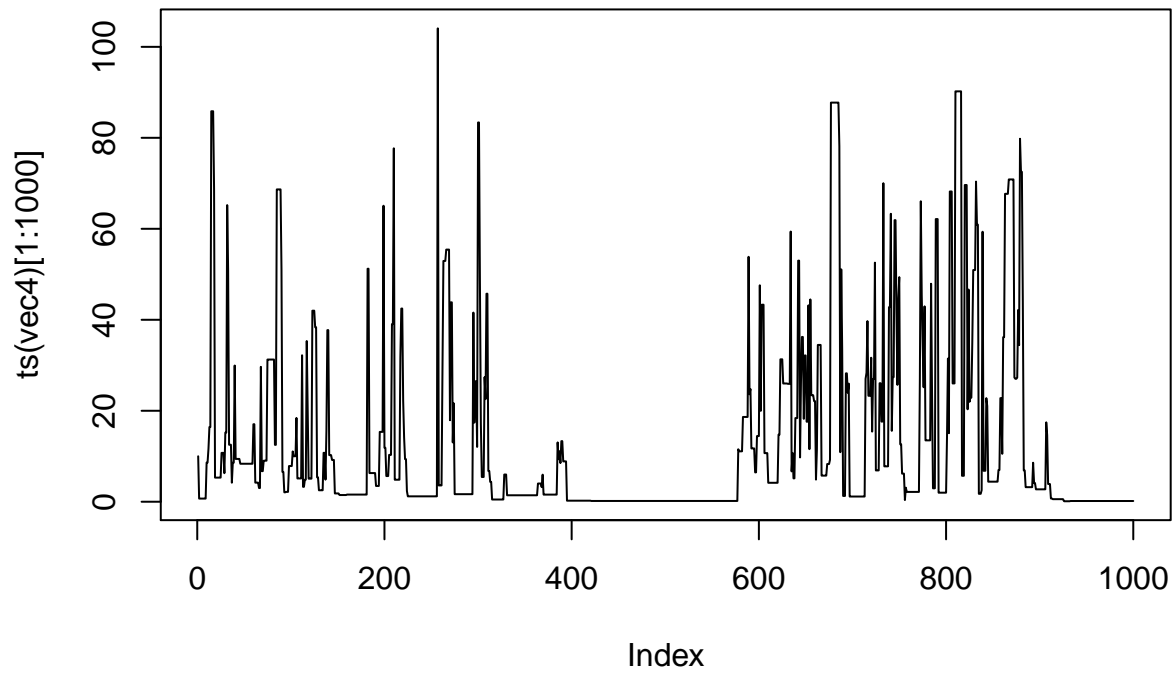
```
plot(ts(vec4),main="vec4")
```

vec4



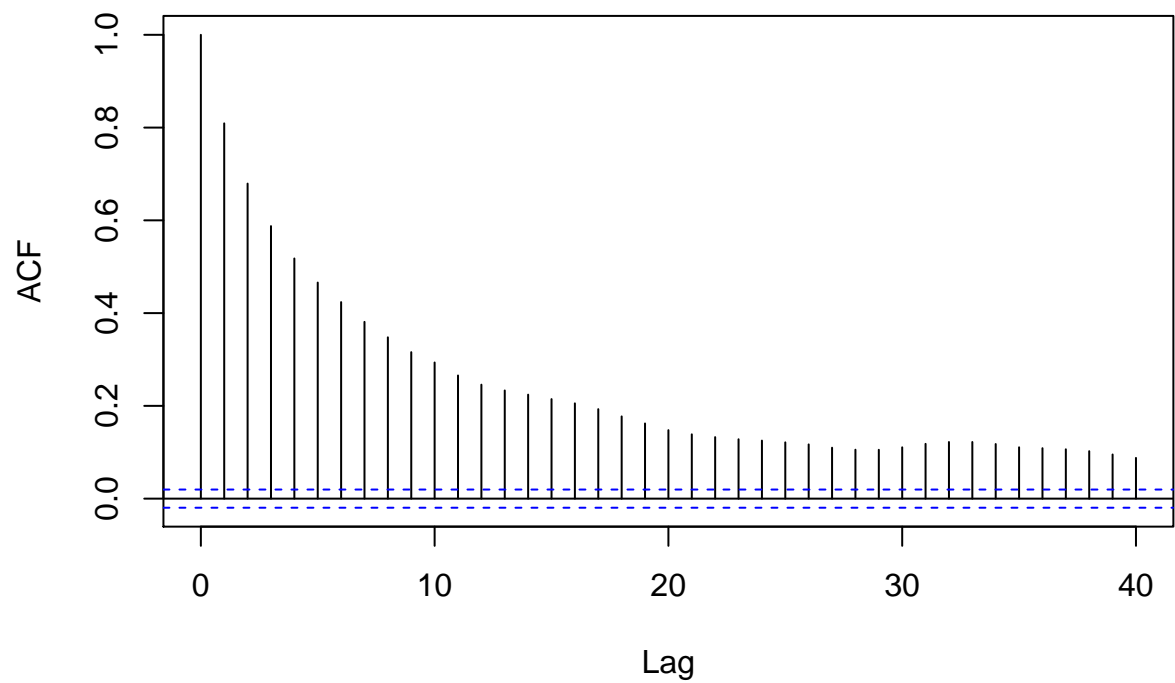
```
plot(ts(vec4)[1:1000],type='l',main="vec4")
```

vec4



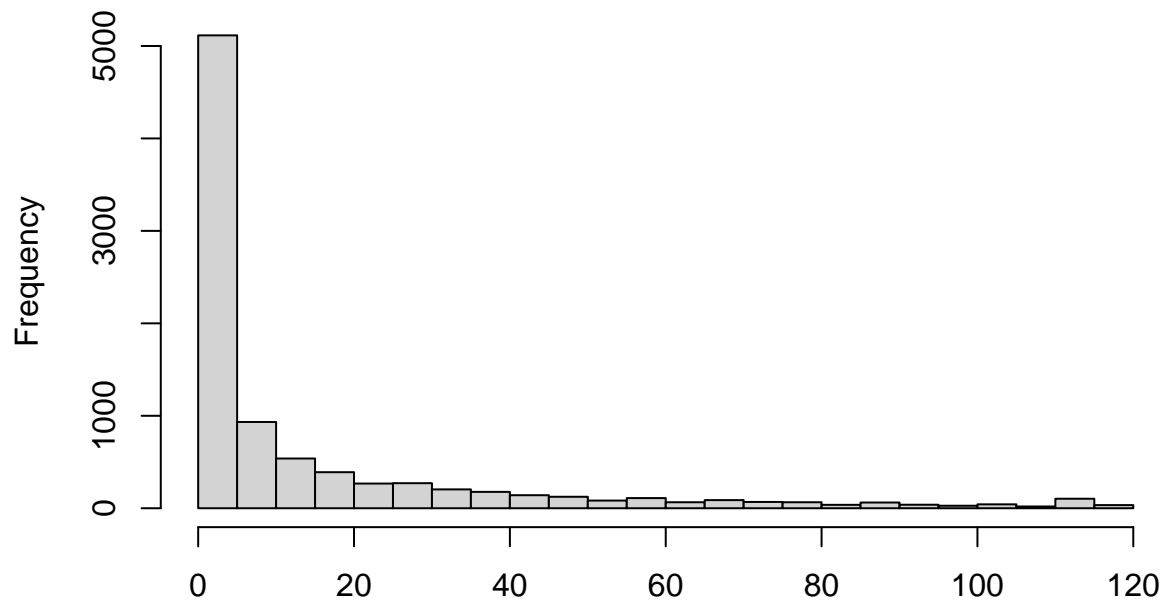
```
acf(vec4)
```

Series vec4



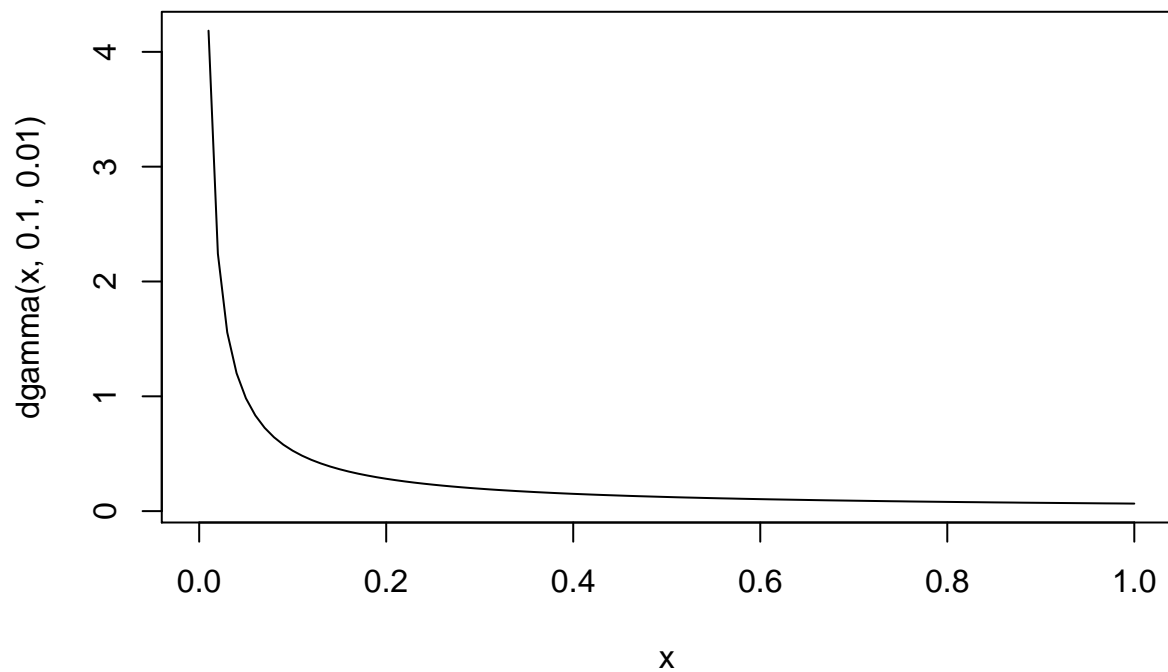
```
hist(vec4[1000:10000],30,main="vec2")
```

vec2



vec4[1000:10000]

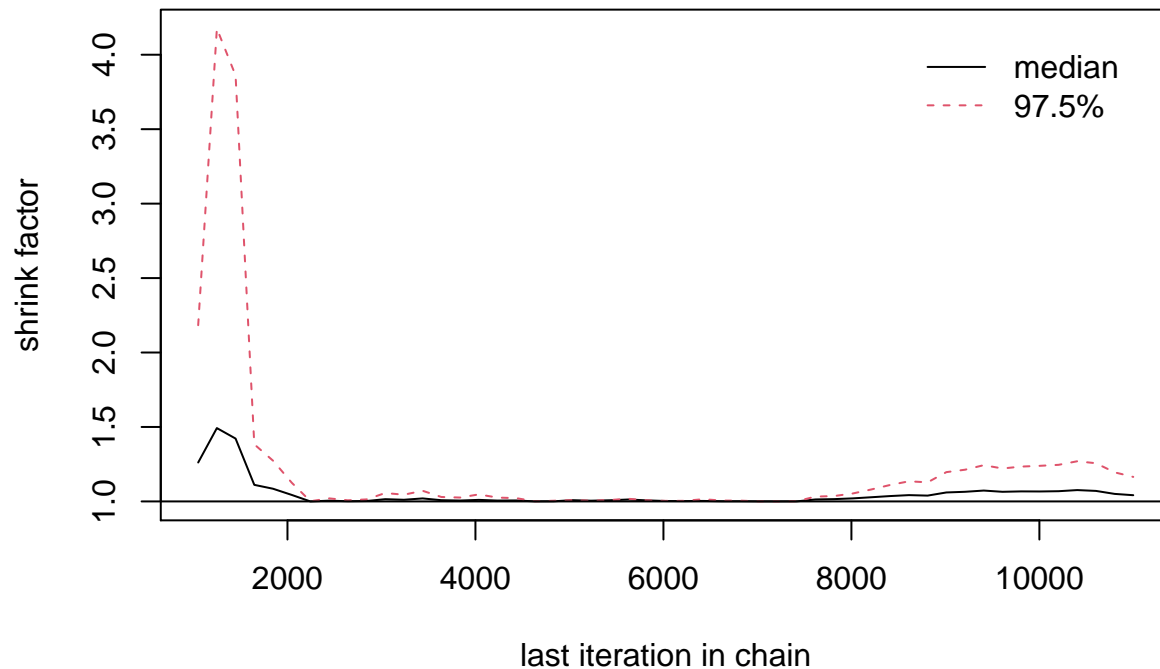
```
curve(dgamma(x,0.1,0.01))
```



```
MCMC3<-mcmc(vec3,start=1000)  
MCMC4<-mcmc(vec4,start=1000)
```

```
# combine different mcmc chain objects to an mcmc list.  
Combined2<-mcmc.list(list(MCMC3,MCMC4))
```

```
# gelman functions are
gelman.plot(Combined2) # for plots
```



```
print(gelman.diag(Combined2)) # for diagnostic values
```

```
## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]      1.04      1.16
```

This one doesn't look so good. It seems to get stuck for long periods of time at low value, and even sometimes at high values.

Your assignment is as follows:

Try out this code for different shape and scale parameters. Notice, as above, how there is a bit of a problem with the sampler getting “stuck” at very small values for some values of a and b . (To see what the distribution should look like use this command: `curve(dgamma(x,0.1,0.01))`).

1. Modify the code to keep track of acceptance probabilities and plot the acf
2. When does the sampling scheme do worst? In what ways is it struggling?
3. Modify this sampling scheme to make it more efficient. (i.e. to remove the issue of it getting stuck at small values.)

Part 3 is quite hard, so you may need to try a few things before you find one that works.