Assignment 6

Jeff Gould

4/22/2021

Theoretical Exercises

1.

$$
\begin{aligned}
y_i &\sim \mathcal{N}\left(\mu, \frac{4\sigma^2}{\alpha_i}\right) \\
\sigma^2 &\sim IG(a, b) \\
\alpha_i &\overset{iid}{\sim} IG(1, 1/2) \\
\pi(\mu) &\propto 1
\end{aligned}
$$

$$
P(\mu, \sigma^2, \alpha | Y) \propto \prod \left[\frac{1}{\sqrt{2\pi \frac{4\sigma^2}{\alpha_i}}} \exp\left(-\frac{(y_i - \mu)^2}{2\frac{4\sigma^2}{\alpha_i}}\right)\right] (\sigma^2)^{-(\alpha+1)} \exp\left(-\frac{b}{\sigma^2}\right) \prod \alpha_i^{-(1+1)} \exp\left(-\frac{1/2}{\alpha_i}\right)
$$

$$
\begin{aligned}
P(\mu | Y, \sigma^2, \alpha) &\propto \prod \frac{1}{\sqrt{2\pi \frac{4\sigma^2}{\alpha_i}}} \exp\left(-\frac{(y_i - \mu)^2}{2\frac{4\sigma^2}{\alpha_i}}\right) \\
&\propto \exp\left(-\frac{1}{2}\sum \frac{(y_i - \mu)^2}{(4\sigma^2/\alpha_i)}\right) \\
&= \exp\left(-\frac{1}{2(4\sigma^2)}\sum \alpha_i (y_i - \bar{y})^2 + \alpha_i n (\bar{y} - \mu)^2\right) \\
&= \exp\left(-\frac{1}{2(4\sigma^2)}\sum \alpha_i (y_i - \bar{y})^2\right) \exp\left(-\frac{1}{2(4\sigma^2)} n (\bar{y} - \mu)^2 \sum \alpha_i\right) \\
&\propto \exp\left(-\frac{1}{2\left(\frac{4\sigma^2}{\sum \alpha_i}\right)}(\mu - \bar{y})^2\right) \\
\mu &\sim \mathcal{N}\left(\bar{y}, \frac{4\sigma^2}{n\sum \alpha_i}\right)
\end{aligned}
$$

$$
\begin{aligned}
P(\alpha_i|Y,\sigma^2,\mu,\alpha_j's) &\propto \frac{1}{\sqrt{2\pi\frac{4\sigma^2}{\alpha_i}}}\exp\left(-\frac{(y_i-\mu)^2}{2\frac{4\sigma^2}{\alpha_i}}\right)\alpha_i^{-(1+1)}\exp\left(-\frac{1/2}{\alpha_i}\right)\\
&= \sqrt{\frac{1/4\sigma^2}{2\pi\alpha_i^3}}\exp\left(-\left(\frac{(y_i-\mu)^2}{2\frac{4\sigma^2}{\alpha_i}}+\frac{1/2}{\alpha_i}\right)\right)\\
&\propto \sqrt{\frac{1}{2\pi\alpha_i^3}}\exp\left(-\frac{1}{2}\frac{\alpha_i^2(y_i-\mu)^2+4\sigma^2}{\alpha_i(4\sigma^2)}\right)\\
&= \sqrt{\frac{1}{2\pi\alpha_i^3}}\exp\left[-\frac{1}{2}\frac{\frac{(y_i-\mu)^2}{4\sigma^2}\alpha_i^2-\frac{y_i-\mu}{\sigma}\alpha_i+1+\frac{y_i-\mu}{\sigma}\alpha_i}{\alpha_i}\right]\\
&= \sqrt{\frac{1}{2\pi\alpha_i^3}}\exp\left[-\frac{1}{2}\left(\frac{\frac{(y_i-\mu)^2}{4\sigma^2}(\alpha_i-\frac{2\sigma}{y_i-\mu})^2}{\alpha_i}+\frac{\frac{y_i-\mu}{\sigma}\alpha_i}{\alpha_i}\right)\right]\\
&= \sqrt{\frac{1}{2\pi\alpha_i^3}}\exp\left[-\frac{1}{2}\frac{(\alpha_i-\frac{2\sigma}{y_i-\mu})^2}{\alpha_i\frac{4\sigma^2}{(y_i-\mu)^2}}-\frac{y_i-\mu}{2\sigma}\right]\\
&\propto \sqrt{\frac{1}{2\pi\alpha_i^3}}\exp\left[-\frac{1}{2}\frac{(\alpha_i-\frac{2\sigma}{y_i-\mu})^2}{\alpha_i\frac{4\sigma^2}{(y_i-\mu)^2}}\right]\\
\alpha_i &\sim InverseGaussian\left(\mu=\frac{2\sigma}{y_i-\mu},\lambda=1\right)
\end{aligned}
$$

$$
\begin{aligned}
P(\sigma^2|Y,\mu,\alpha) &\propto \left(\prod\frac{1}{\sqrt{2\pi(4\sigma^2/\alpha_i)}}\exp\left(-\frac{(y_i-\mu)^2}{2(4\sigma^2/\alpha_i)}\right)\right)(\sigma^2)^{-(a+1)}\exp\left(-\frac{b}{\sigma^2}\right)\\
&\propto (\sigma^2)^{-n/2}(\sigma^2)^{-(a+1)}\exp\left(-\frac{1}{2(4\sigma^2)}\sum\alpha_i(y_i-\mu)^2\right)\exp\left(-\frac{b}{\sigma^2}\right)\\
&\propto (\sigma^2)^{-(a+n/2+1)}\exp\left(-\frac{1/8\sum\alpha_i(y_i-\mu)^2+b}{\sigma^2}\right)\\
\sigma^2 &\sim IG\left(a+n/2, b+\frac{1}{8}\sum\alpha_i(y_i-\mu)^2\right)
\end{aligned}
$$

Pseudo Code Gibbs Sampler:

- Initialize vectors for `mu`, `sigma2`, and all the `alpha_i` (or perhaps a matrix `alpha`)
- Create initial values for each parameter

for b in 2:B :

Sample `mu[b]` from `rnorm(mean(Y), 4*sigma2[b-1] / n * sum(alpha_i[b-1]))`

Sample `sigma[b]` from `rinvgamma(a + n/2, b + 1/8 sum(alpha_i(y_i - mu[b-1])^2))`

Sample each `alpha_i[b]` from the inverse Gaussian distribution, with parameters $\lambda = 1$ and $\mu = 2\sigma[b-1]/(y_i-\mu[b-1])$

2.

$$
\begin{aligned}
P(x) &= \binom{n}{x} p^x (1-p)^{n-x} \\
p &\sim beta(\alpha, \beta) \\
\alpha &\sim Gamma(a_1, b_1) \\
\beta &\sim Gamma(a_2, b_2)
\end{aligned}
$$

$$
\begin{aligned}
P(p, \alpha, \beta | X) &= \mathcal{L}(x) P(p|\alpha, \beta) P(\alpha, \beta) \\
&\propto \binom{n}{x} p^x (1-p)^{n-x} \frac{p^{\alpha-1}(1-p)^{\beta-1}}{B(\alpha, \beta)} \alpha^{a_1 - 1} \exp(-b_1 \alpha) \beta^{a_2-1} \exp(-b_2 \beta)
\end{aligned}
$$

$$
\begin{aligned}
P(p|x, \alpha, \beta) &\propto p^{x+\alpha-1}(1-p)^{n+\beta-x-1} \\
&\sim beta(x+\alpha, n+\beta-x)
\end{aligned}
$$

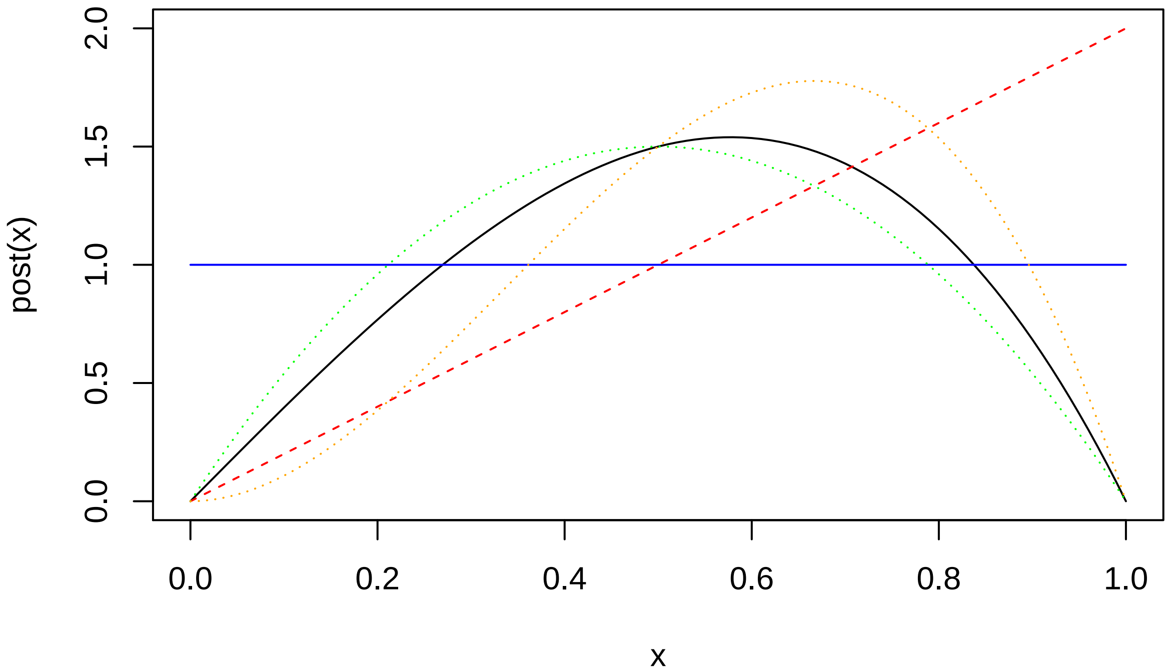
$$
\begin{aligned}
P(\alpha|\beta, x, p) &\propto p^{\alpha-1} \alpha^{a_1-1} \exp(-b_1\alpha)/B(\alpha, \beta) \\
&\propto \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)} p^{\alpha-1} \alpha^{a_1-1} \exp(-b_1 \alpha) \\
P(\beta|\alpha, x, p) &\propto (1-p)^{\beta-1} \beta^{a_2-1} \exp(-b_2\beta)/B(\alpha, \beta) \\
&\propto \frac{\Gamma(\alpha+\beta)}{\Gamma(\beta)} (1-p)^{\beta-1} \beta^{a_2-1} \exp(-b_2\beta)
\end{aligned}
$$

For α and β, we may want to use the Gamma distribution as our proposal distribution, as there is still the kernel for a gamma distribution from out prior

Computing Exercises

$$p(\theta) = ab\theta^{a-1}(1-\theta)^{b-1}$$

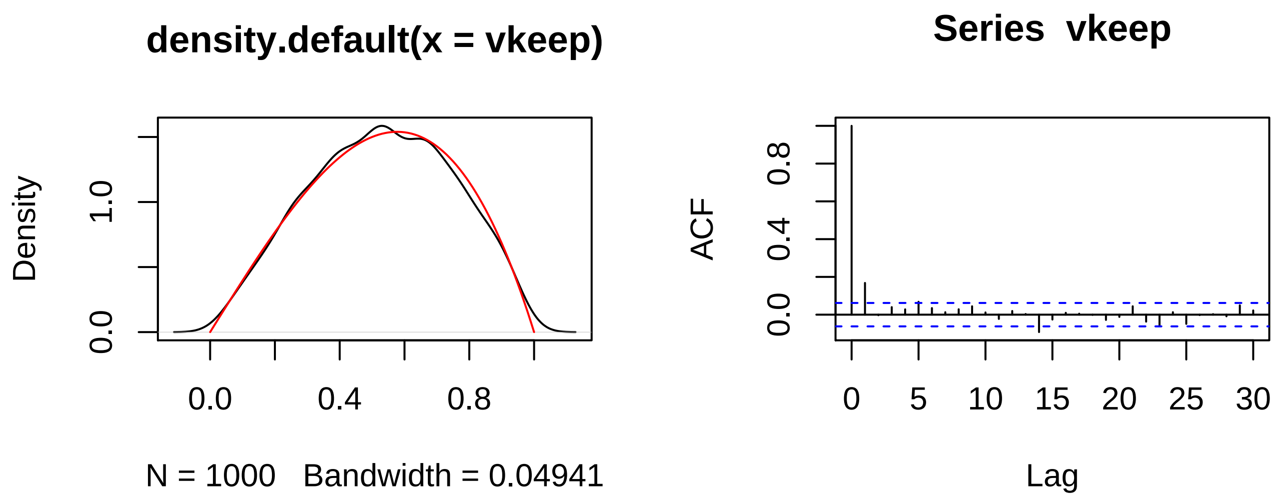
```r
post <- function(theta, a = 2, b = 2){
  a * b * theta^(a-1) * (1 - theta^a)^(b-1)
}

curve(post, from = 0, to = 1, ylim = c(0, 2))
curve(dbeta(x, 1, 1), from = 0, to = 1, add = TRUE, col= 'blue')
curve(dbeta(x, 2, 1), from = 0, to = 1, add = TRUE, col= 'red', lty = 2)
curve(dbeta(x, 2, 2), from = 0, to = 1, add = TRUE, col= 'green', lty = 3)
curve(dbeta(x, 3, 2), from = 0, to = 1, add = TRUE, col= 'orange', lty = 3)
```




```r
MH_algo <- function(B, alpha, beta){
  set.seed(1218)
  vec       <- vector("numeric", B)
  vec[1]    <- 0.5
  x <- 0.5
  ar        <- vector("numeric", B)
  for (i in 2:B) {
    can     <- rbeta(1, alpha, beta)
    r     <- (post(can)/dbeta(can, alpha, beta)) / (post(x)/dbeta(x, alpha, beta))
    u       <- runif(1)
    if (u < min(r,1)){
      x         <- can
      ar[i] <- 1
    }
    vec[i]  <- x
  }

  vkeep   <- vec[-(1:(B/2))]
  arkeep  <- ar[-(1:(B/2))]

  print(glue::glue("Acceptance Ratio: {mean(arkeep)}"))
  plot(density(vkeep))
  curve(post, from = 0, to = 1, add = TRUE, col = "red")
  acf(vkeep)
  return(vkeep)
}
```

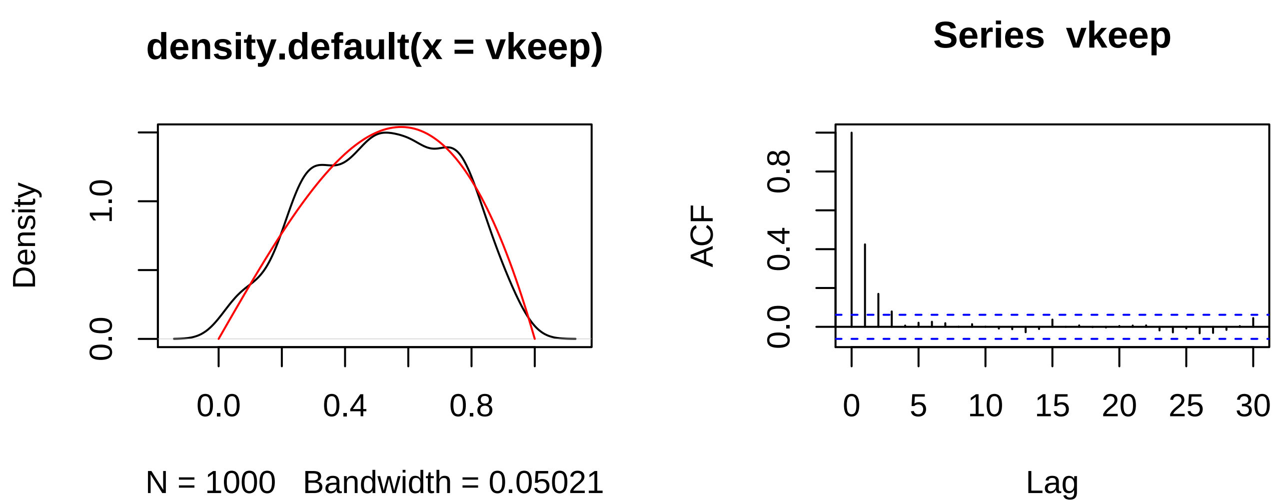
```r
beta_1_1 <- MH_algo(2000, 1, 1)
```

```
## Acceptance Ratio: 0.736
```



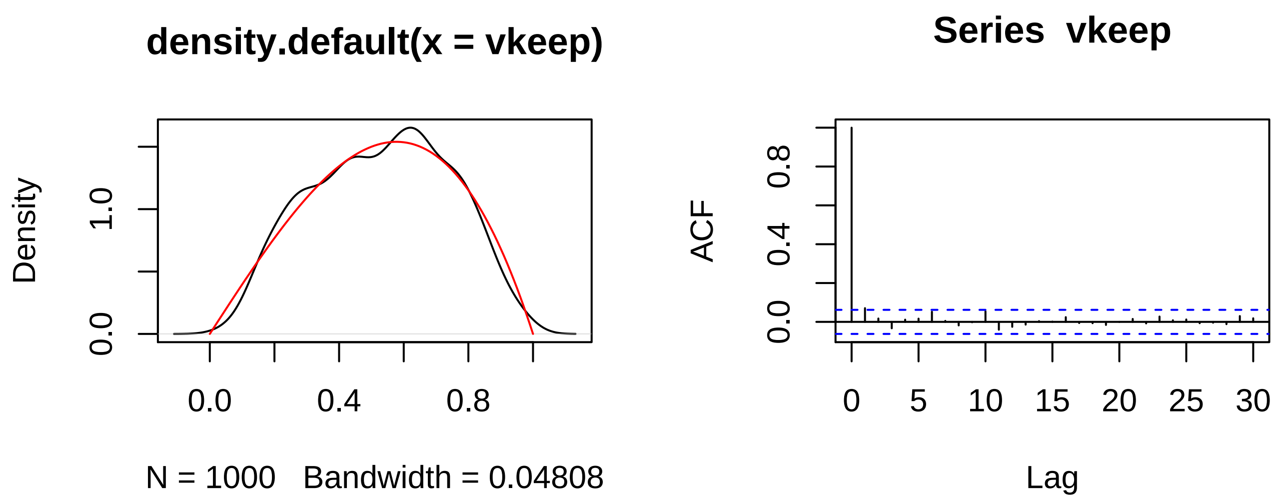

```r
beta_2_1 <- MH_algo(2000, 2, 1)
```

```
## Acceptance Ratio: 0.672
```



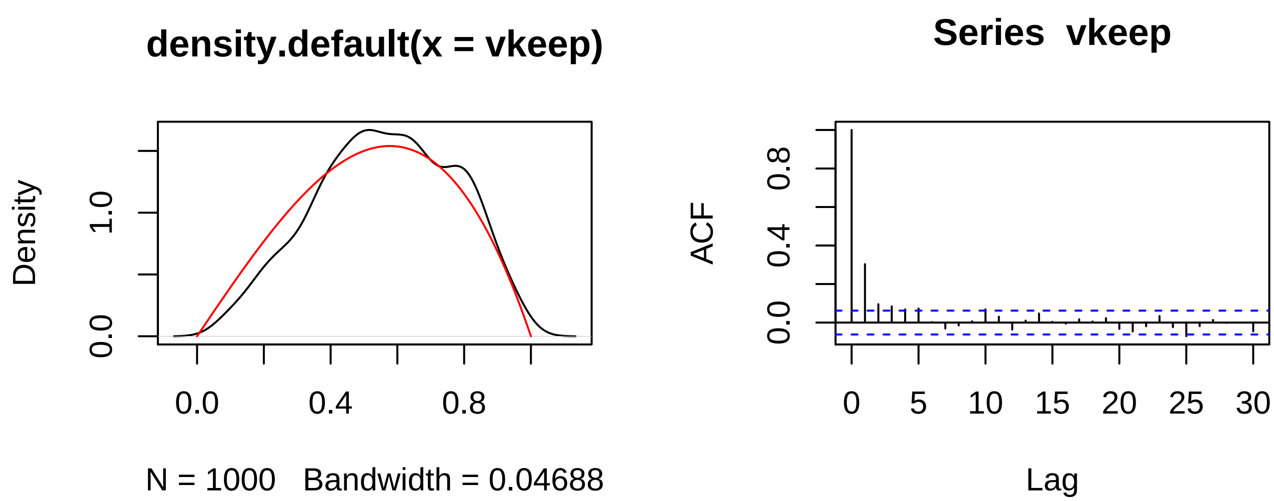

```r
beta_2_2 <- MH_algo(2000, 2, 2)
```

```
## Acceptance Ratio: 0.913
```




```r
beta_3_2 <- MH_algo(2000, 3, 2)
```

```
## Acceptance Ratio: 0.869
```



The $Beta(1, 1)$ seems most ideal, as it has very little auto-correlation, the sampled density closely approximates the underlying distribution, and the acceptance rate is a good 73.6%, compared to as much as 91.3% for the $Beta(2, 2)$ proposal density

With no thinning, the auto-corellation plots are mostly good. There is some slight auto-correlation between the sample immediately following, but by the third it disappears. We may want to thin and take every other sample due to this.

Analysis Exercises

1

We find that the model predicts the underlying distribution pretty accurately, though perhaps slightly shifted to the right. It is tough to say whether we have a slight shift in our sample distributions or whether it's due to the kernel estimation with the drop-off after 0. We also see that we have convergence with the Geweke Diagnositc values around 0.55 and 0.05

```
set.seed(1789)
n <- 100
x <- rbinom(n, 1, 0.27)
X <- sum(x)

a1 <- b1 <- a2 <- b2 <- 1

B           <- 2*10000
p         <- vector("numeric", B)
alpha         <- vector("numeric", B)
alpha[1] <- a <- 1

arAlpha <- vector("numeric", B)

beta          <- vector("numeric", B)
beta[1] <- b <- 1

arBeta <- vector("numeric", B)

p[1]    <- mean(x)

n           <- length(x)

p_alpha <- function(.alpha, .beta, .p, .a1 = 1, .b1 = 1){
  (gamma(.alpha + .beta) / gamma(.alpha)) * (.p)^(.alpha - 1) * .alpha ^(.a1 - 1) * exp(-.b1*.alpha)
}



p_beta <- function(.beta, .alpha, .p, .a2 = 1, .b2 = 1){
  (gamma(.alpha + .beta) / gamma(.beta)) * (1 - .p)^(.beta - 1) * .beta ^(.a2 - 1) * exp(-.b2*.beta)
}

for(b in 2:B){

  ### sample P ###
  p[b]  <- rbeta(1, shape1 = X + alpha[b-1], shape2 = n + beta[b-1] - X)

  ### sample alpha, beta ###
  alphastar <- rgamma(1, a1, b1)
  betastar <- rgamma(1, a2, b2)

  r1       <- (p_alpha(alphastar, .beta = beta[b-1], .p = p[b-1]) / (dgamma(alphastar, a1, b1 ))) /
    (p_alpha(alpha[b-1], .beta = beta[b-1], .p = p[b-1]) / dgamma(alpha[b-1], a1, b1))

  r2       <- (p_beta(betastar, .alpha = alpha[b-1], .p = p[b-1]) / dgamma(betastar, a2, b2)) /
    (p_beta(beta[b-1], .alpha = alpha[b-1], .p = p[b-1]) / dgamma(beta[b-1], a1, b1))
```

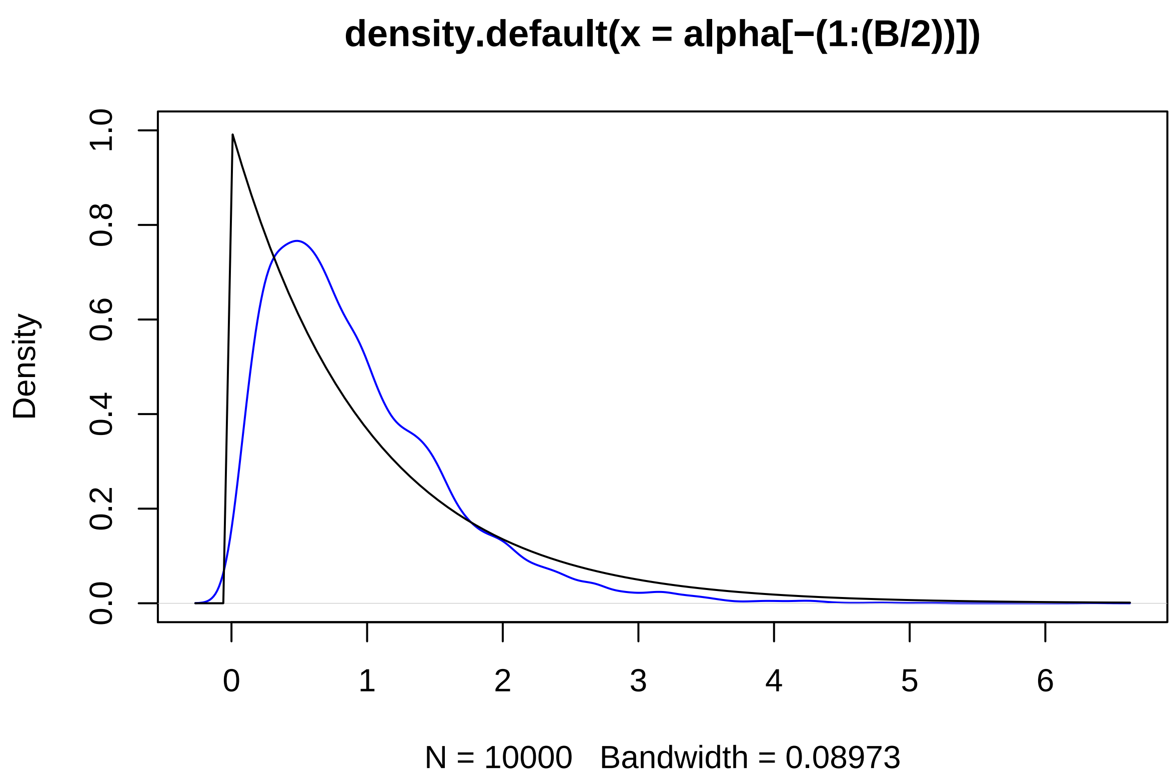
```r
  U      <- runif(1)
  if(U < min(r1,1)){
    a        <- alphastar
    arAlpha[b]  <- 1
  }
  alpha[b]      <- a

  if(U < min(r2,1)){
    bb       <- betastar
    arBeta[b]   <- 1
  }
  beta[b]       <- bb

}

mean(arAlpha)
```

```
## [1] 0.73475
```

```r
mean(arBeta)
```

```
## [1] 0.70135
```

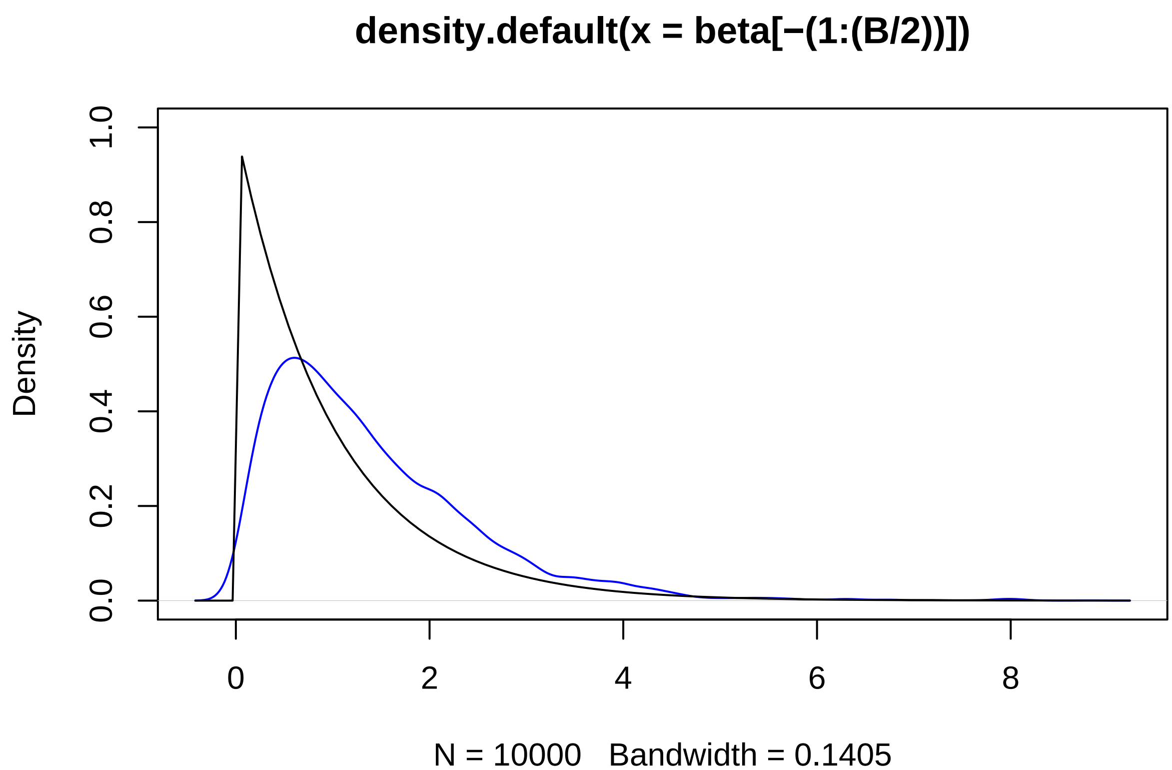
```r
plot(density(alpha[-(1:(B/2))]), col = "blue", ylim = c(0,1))
curve(dgamma(x, 1, 1), add = T)
```



```r
plot(density(beta[-(1:(B/2))]), col = "blue", ylim = c(0,1))
curve(dgamma(x, 1, 1), add = T)

mean(p)
```

```
## [1] 0.272917
```

```r
coda::geweke.diag(beta[-(1:(B/2))])
```




```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##   var1
## 0.5512
```

```r
coda::geweke.diag(alpha[-(1:(B/2))])
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##    var1
## 0.05443
```

2

$$P(\lambda, \mu | s_i) \propto \lambda^{-1} \prod \left(\frac{\lambda}{2\pi s_i^3} \right)^{1/2} \exp\left[-\frac{\lambda (s_i - \mu)^2}{2\mu^2 s_i} \right]$$

$$\begin{aligned} P(\lambda | \mu, s_i) &\propto \lambda^{-1} \lambda^{n/2} \exp\left[-\sum \frac{\lambda (s_i - \mu)^2}{2\mu^2 s_i} \right] \\ &= \lambda^{n/2-1} \exp\left[-\lambda \sum \frac{(s_i - \mu)^2}{2\mu^2 s_i} \right] \\ \lambda &\sim Gamma\left(\frac{n}{2}, \frac{(s_i - \mu)^2}{2\mu^2 s_i} \right) \end{aligned}$$

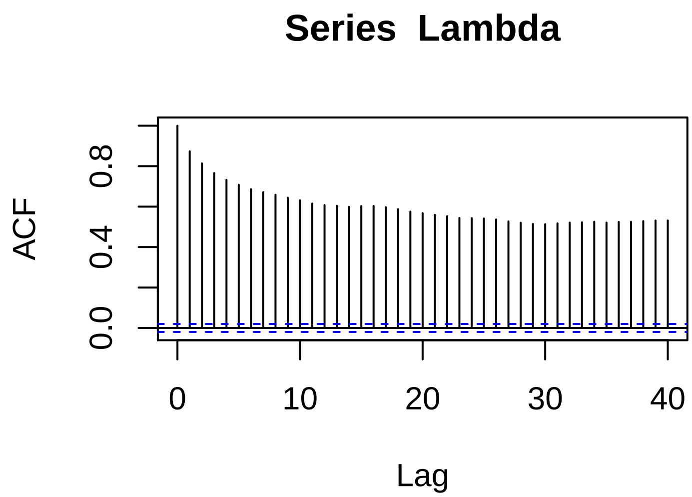
$$P(\mu | \lambda, s_i) \propto \exp\left(-\lambda \sum \frac{(s_i - \mu)^2}{2\mu^2 s_i} \right)$$

```
S <- read.table("coupsd.txt")[,1]

B <- 20000

lambda <- vector("numeric", B)
lambda[1] <- 1 / sd(S)

mu <- vector("numeric", B)
m <- mean(S)
mu[1] <- m

ar <- vector("numeric", B)

P_mu <- function(.mu, .lambda){
  -.lambda/(2 * .mu^2) * sum( (S - .mu)^2 / S )
}

set.seed(1980)
for (b in 2:B) {

  lambda[b] <- rgamma(1,
                      shape = length(S)/2,
                      rate = sum( (S - mu[b-1])^2 / (2 * mu[b-1]^2 * S))
                      )

  mu_star <- rnorm(1, mean(S), 0.185)

  U <- runif(1)

  r <- P_mu(.mu = mu_star, .lambda = lambda[b-1]) / P_mu(.mu = mu[b-1], .lambda = lambda[b-1])

  if(U < r){
    m <- mu_star
    ar[b] <- 1
  }

  mu[b] <- m
```

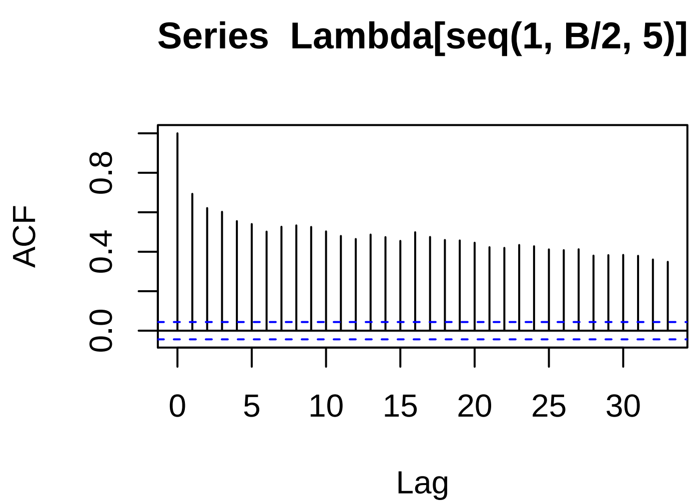
```r
}
mean(ar[B/2:B])
```

```
## [1] 0.4479224
```

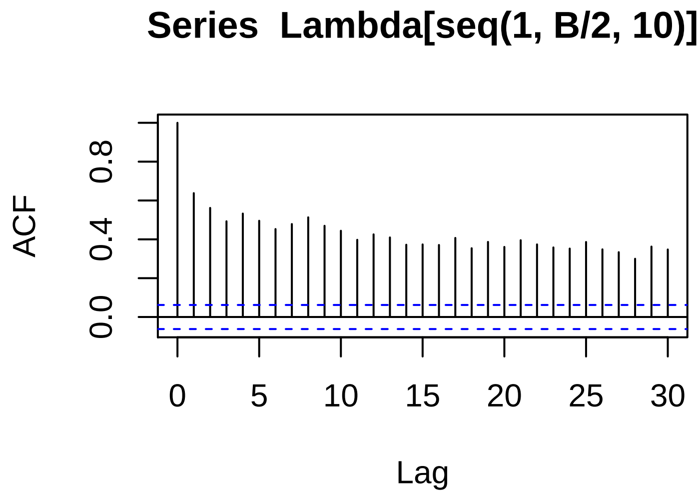
```r
Lambda <- lambda[(B/2+1):B]
acf(Lambda)
```



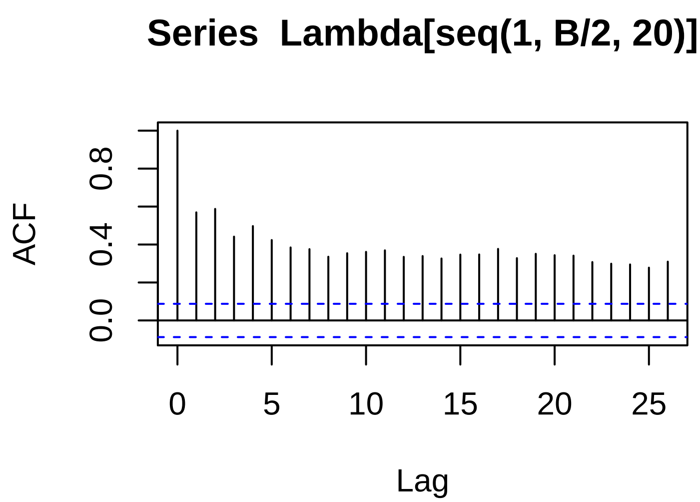

```r
acf(Lambda[seq(1,B/2, 5)])
```



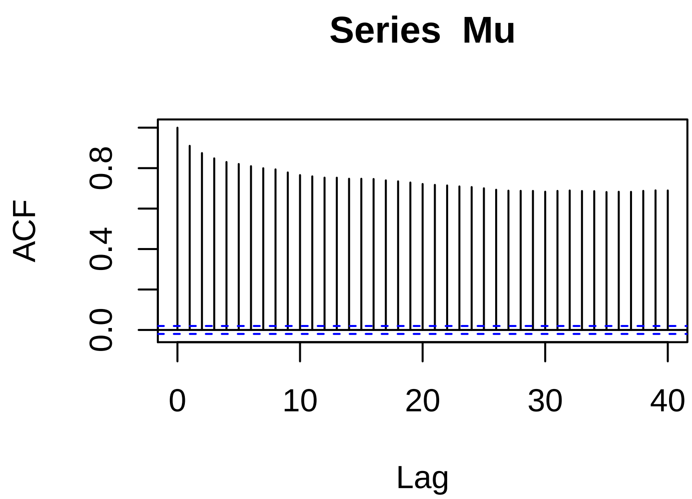
```r
acf(Lambda[seq(1,B/2, 10)])
```



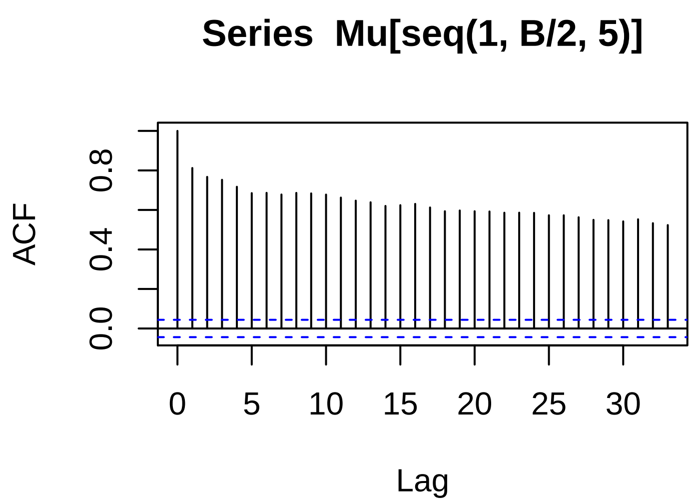

```r
acf(Lambda[seq(1,B/2, 20)])
```



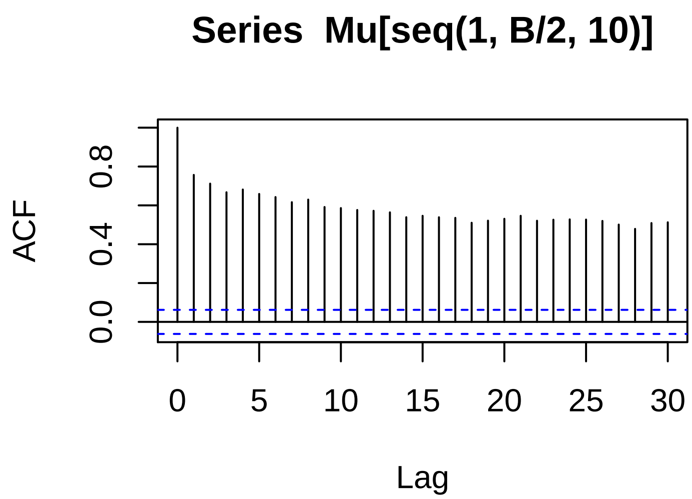

```r
Mu <- mu[(B/2+1):B]
acf(Mu)
```



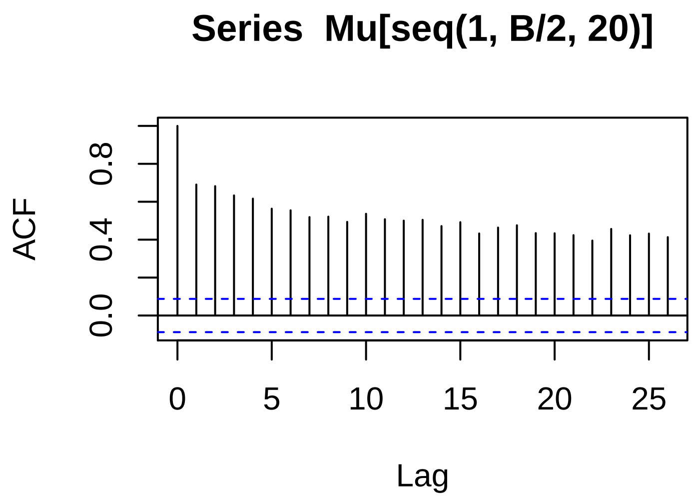

```r
acf(Mu[seq(1,B/2, 5)])
```



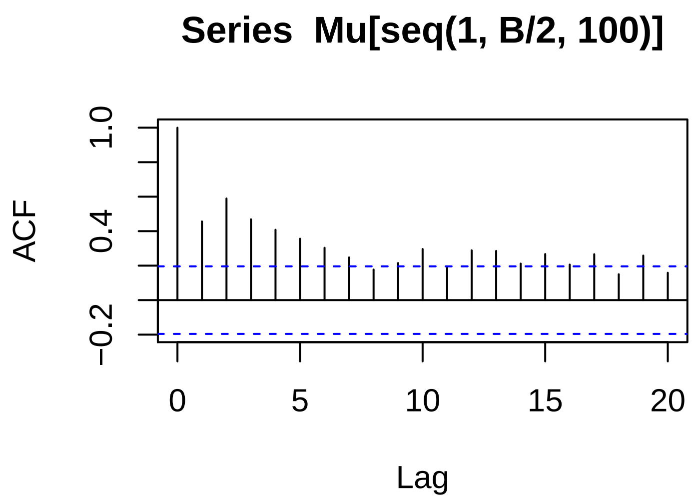

```r
acf(Mu[seq(1,B/2, 10)])
```




```r
acf(Mu[seq(1,B/2, 20)])
```




```r
acf(Mu[seq(1,B/2, 100)])
```



We find that the autocorrelation decreases each time we thin, but even at taking one out of every 20 samples there is still a high degree of autocorrelatoin between samples. If we increase the thinning up to 100, then most of the autocorrelation for `Lambda` disappears, and while better it still exists for `Mu`. When thinning by 100, we are then down to just 100 samples