STAT 221 - ASSIGNMENT 4

GREG TAM, STUDENT ID: 70908239

1.1 First, we reparameterize μ as $\lambda = \theta \cdot \mu$ so that

$$\begin{aligned}\mathbb{E}[Y_i|\theta,\mu] &= \mathbb{E}[\mathbb{E}[Y_i|\theta,\mu,N]] \\ &= \mathbb{E}[N\theta] \\ &= \theta\cdot\mu \\ &= \lambda\end{aligned}$$

The suggested distribution we are given is $\mathbb{P}(\lambda,\theta) \propto \lambda^{-1}$. We do a change of variables from (λ,θ) to (μ,θ). Define g_1, g_2 as

$$\begin{aligned}g_1(\lambda,\theta) &= \frac{\lambda}{\theta} = \mu \\ g_2(\lambda,\theta) &= \theta\end{aligned}$$

and their inverses, h_1, h_2, are

$$\begin{aligned}h_1(\mu,\theta) &= \theta\cdot\mu = \lambda \\ h_2(\mu,\theta) &= \theta\end{aligned}$$

respectively. Now, this gives us our Jacobian for the transformation

$$|J| = \begin{vmatrix}\frac{\partial h_1}{\partial\mu} & \frac{\partial h_1}{\partial\theta} \\ \frac{\partial h_2}{\partial\mu} & \frac{\partial h_2}{\partial\theta}\end{vmatrix} = \begin{vmatrix}\theta & \mu \\ 0 & 1\end{vmatrix} = \theta$$

So our prior distribution on (μ,θ) is

$$\begin{aligned}\mathbb{P}(\mu,\theta) &= \mathbb{P}(h_1(\mu,\theta),h_2(\mu,\theta))\,|J| \\ &= \mathbb{P}(\mu\theta,\theta)\,\theta \\ &= \frac{1}{\mu\theta}\theta\mathbb{1}_{\theta\in[0,1]} \\ &\propto \frac{1}{\mu}\mathbb{1}_{\theta\in[0,1]}\end{aligned}$$

Using this, we have that

$$\begin{aligned}\mathbb{P}(N,\theta) &= \int \mathbb{P}(N,\theta,\mu)\ \mathrm{d}\mu \\ &= \int_0^\infty \mathbb{P}(N|\theta,\mu)\,\mathbb{P}(\mu,\theta)\ \mathrm{d}\mu \\ &\propto \int \frac{\mu^N}{N!}e^{-\mu}\frac{1}{\mu}\mathbb{1}_{\theta\in[0,1]}\,\mathrm{d}\mu \\ &= \frac{1}{N!}\mathbb{1}_{\theta\in[0,1]}\int_0^\infty \mu^{N-1}e^{-\mu}\,\mathrm{d}\mu \\ &= \frac{\Gamma(N)}{N!}\mathbb{1}_{\theta\in[0,1]} \\ &= \frac{1}{N}\mathbb{1}_{\theta\in[0,1]}\end{aligned}$$

This is sensible as the probability of sampling N is less likely the larger N is and more likely the smaller N is.

1.2 The support of λ is $(0,\infty)$ and the support of θ is $[0,1]$. If we take

$$\begin{aligned}\int_0^\infty\int_0^1 \mathbb{P}(\lambda,\theta)\ \mathrm{d}\theta\mathrm{d}\lambda &= \int_0^\infty\int_0^1 \frac{1}{\lambda}\,\mathrm{d}\theta\mathrm{d}\lambda \\ &= \infty\end{aligned}$$

and because this is finite, we cannot get a normalizing constant, so this prior is not proper.

1.3 We have that $Y_1, \ldots, Y_n \sim \text{Bin}(N, \theta)$, so that

$$\mathbb{P}(Y_i = y_i | N, \theta) = \binom{N}{y_i} \theta^N (1-\theta)^{N-y_i}$$

To get $\mathbb{P}(Y_i | \mu, \theta)$, we take

$$\begin{aligned}
\mathbb{P}(Y_i = y_i | \mu, \theta) &= \sum_{n=y_i}^{\infty} \mathbb{P}(Y_i = y_i, N = n | \mu, \theta) \\
&= \sum_{n=y_i}^{\infty} \mathbb{P}(Y_i = y_i | N = n, \mu, \theta) \, \mathbb{P}(N = n | \mu, \theta) \\
&= \sum_{n=y_i}^{\infty} \mathbb{P}(Y_i = y_i | N = n, \theta) \, \mathbb{P}(N = n | \mu, \theta) \\
&= \sum_{n=y_i}^{\infty} \binom{n}{y_i} \theta^{y_i} (1-\theta)^{n-y_i} \frac{\mu^n}{n!} e^{-\mu} \\
&= \sum_{n=y_i}^{\infty} \frac{1}{y_I! (n-y_i)!} \theta^{y_i} (1-\theta)^{n-y_i} \mu^n e^{-\mu} \\
&= \frac{(\theta\mu)^{y_i} e^{-\mu}}{y_i!} \sum_{n=y_i}^{\infty} \frac{(1-\theta)^{n-y_i}}{(n-y_i)!} \mu^{n-y_i}
\end{aligned}$$

Now, reparameterizing this buy letting $t = n - y_i$, we get

$$\begin{aligned}
\mathbb{P}(Y_i = y_i | \mu, \theta) &= \frac{(\theta\mu)^{y)i} e^{-\mu}}{y_i!} \sum_{t=0}^{\infty} \frac{\{(1-\theta)\mu\}^t}{t!} \\
&= \frac{(\theta\mu)^{y)i} e^{-\mu}}{y_i!} e^{(1-\theta)\mu} \\
&= \frac{(\theta\mu)^{y_i} e^{-\theta\mu}}{y_i!}
\end{aligned}$$

and so we have

$$Y_i | \mu, \theta \sim \text{Pois}(\theta\mu) \sim \text{Pois}(\lambda)$$

This gives us log-likelihood

$$\ell = \log \mathbb{P}(Y_i = y | \lambda, \theta) = y_i \log \lambda - \lambda - \log(y_i!)$$

Now, we take the partial derivatives of the log-likelihood with respect to λ and θ, which gives

$$\begin{aligned}
\ell_\lambda &= \frac{y_i}{\lambda} - 1 \\
\ell_\theta &= \frac{\partial \ell}{\partial \lambda} \frac{\partial \lambda}{\partial \theta} \\
&= \left(\frac{y_i}{\lambda} - 1 \right) \mu \\
&= \frac{y_i}{\theta} - \mu
\end{aligned}$$

Taking the second derivatives, we get

$$\begin{aligned}
\ell_{\lambda\lambda} &= -\frac{y_i}{\lambda^2} \\
\ell_{\theta\theta} &= -\frac{y_i}{\theta^2} \\
\ell_{\lambda\theta} &= -\frac{y_i}{\lambda^2} \mu \\
&= -\frac{y_i}{(\mu\theta)^2} \mu \\
&= -\frac{y_i}{\lambda\theta}
\end{aligned}$$

Taking the expectation multiplied by -1 gives

$$\begin{aligned}
-\mathbb{E}[\ell_{\lambda\lambda}] &= -\mathbb{E}\left[\frac{y_i}{\lambda^2} \right] = \frac{\lambda\theta}{\lambda^2} = \frac{\theta}{\lambda} \\
-\mathbb{E}[\ell_{\theta\theta}] &= -\mathbb{E}\left[\frac{y_i}{\theta^2} \right] = \frac{\lambda\theta}{\theta^2} = \frac{\lambda}{\theta}
\end{aligned}$$

$$-\mathbb{E}[\ell_{\lambda\theta}] = -\mathbb{E}\Big[\frac{y_i}{\lambda\theta}\Big] - \frac{\lambda\theta}{\lambda\theta} = 1$$

so we have

$$\mathcal{I}(\lambda,\theta) = \begin{pmatrix} \theta/\lambda & 1 \\ 1 & \lambda/\theta \end{pmatrix}$$

which gives the Jeffrey's prior as

$$\begin{aligned}\mathbb{P}(\lambda,\theta) \propto \sqrt{|\mathcal{I}(\lambda,\theta)|} &\propto \sqrt{\left|\frac{\theta}{\lambda}\frac{\lambda}{\theta} - 1^2\right|} \\ &\propto \sqrt{|0|} \\ &\propto 0\end{aligned}$$

This is clearly a degenerate prior and so $\mathbb{P}(\lambda,\theta)$ is informative in the sense of Jeffrey's.

1.4 First, we derive the posterior distribution of (N,θ) from which we will sample from. From previous parts, we have that

$$\begin{aligned}\mathbb{P}(N,\theta) &\propto \frac{1}{N}\mathbb{1}_{\theta\in[0,1]} \\ \mathbb{P}(Y_1,\ldots,Y_n|N,\theta) &= \prod_{i=1}^{n}\binom{N}{y_i}\theta^{y_i}(1-\theta)^{N-y_i}\end{aligned}$$

Using this, we can get the posterior distribution:

$$\begin{aligned}\mathbb{P}(N,\theta|Y_1,\ldots,Y_n) &\propto \mathbb{P}(Y_1,\ldots,Y_n|N,\theta)\,\mathbb{P}(N,\theta) \\ &= \left\{\prod_{i=1}^{n}\binom{N}{y_i}\theta^{y_i}(1-\theta)^{N-y_i}\right\}\frac{1}{N}\mathbb{1}_{\theta\in[0,1]} \\ &= \left\{\prod_{i=1}^{n}\binom{N}{y_i}\right\}\frac{1}{N}\theta^{S}(1-\theta)^{nN-S}\mathbb{1}_{\theta\in[0,1]}\end{aligned}$$

where $S = \sum_{i=1}^{n} y_i$.

To do our Metropolis-Hastings on this, we initialize N as $\max(y)$ and θ as 0.5. Then at each iteration t, we sample $N_t|Y_1,\ldots,Y_n$ from a geometric. We wish that the mean of this is equal to N_{t-1}, that is

$$\mathbb{E}[N_t] = \frac{1-p}{p} = N_{t-1}$$

We achieve this by setting $p = \frac{1}{1+N_{t+1}}$, so we have

$$N_t \sim \text{Geom}\left(\frac{1}{1+N_t}\right)$$

Next, we sample $\theta_t|N_t,Y_1,\ldots,Y_n$. To do this we look at the posterior probability of (N,θ) given $Y_1,\ldots,Y_n$:

$$\begin{aligned}\mathbb{P}(\theta|N,Y_1,\ldots,Y_n) &\propto \mathbb{P}(N,\theta|Y_1,\ldots,Y_n) \\ &\propto \theta^{S}(1-\theta)^{nN-S}\mathbb{1}_{\theta\in[0,1]}\end{aligned}$$

This tells us that

$$\theta|N,Y_1,\ldots,Y_n \sim \text{Beta}(S+1, nN-S+1)$$

where $S = \sum_{i=1}^{n} Y_i$ as before. The distribution of (N,θ) falls mostly on a asymptote, if we reparameterize it to (M,θ) where $M = N\theta$, then the distribution of (M,θ) covers a thick band. This is due to the fact that M and θ are less correlated than N and θ.

Therefore, in our iterations, at step t, we are given $\theta_{t-1}, N_{t-1}, M_{t-1}, N_t$. We can sample

$$M_t \sim N_{t-1}\cdot\text{Beta}(S+1, nN_{t-1}-S+1)$$

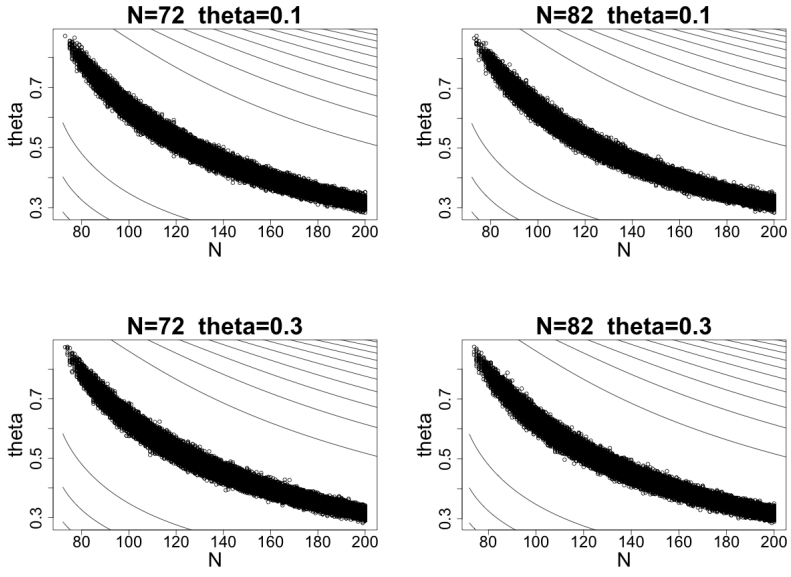
which is a Beta random variable scaled up by N_{t-1}. We can simply rescale this by dividing by N_t, which gives

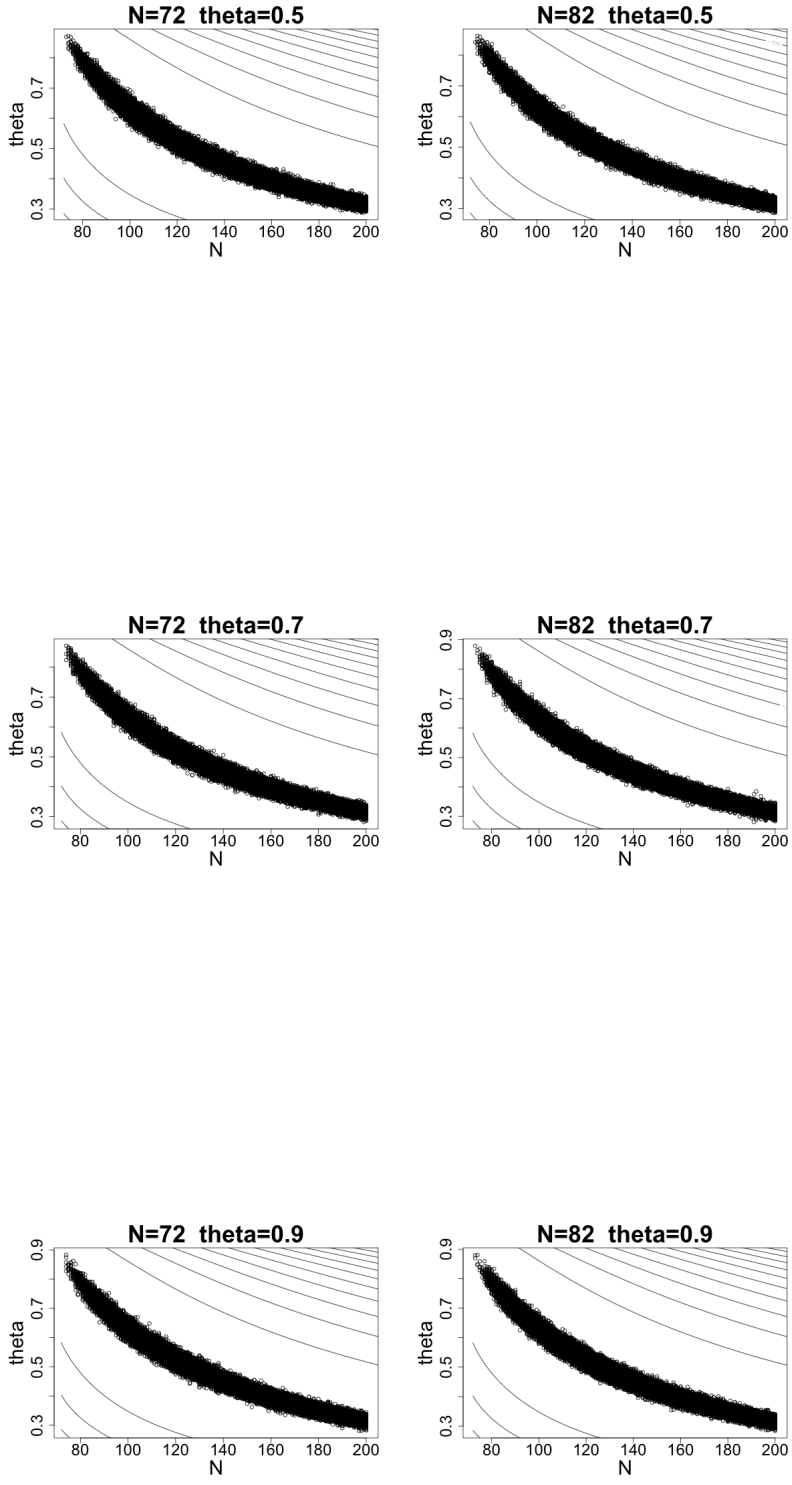
$$\theta_t = \frac{M_t}{N_t}$$

Since N is inversely proportional to θ, small values of θ can lead to large values of N. We put a limit on how large N can be (500 for waterbuck and 200 for impala) so our samples are not too wild. Similarly, we ensure N cannot be below $\max(y)$ or else the Binomial distribution is not defined and θ must be between 0 and 1. The code is below:

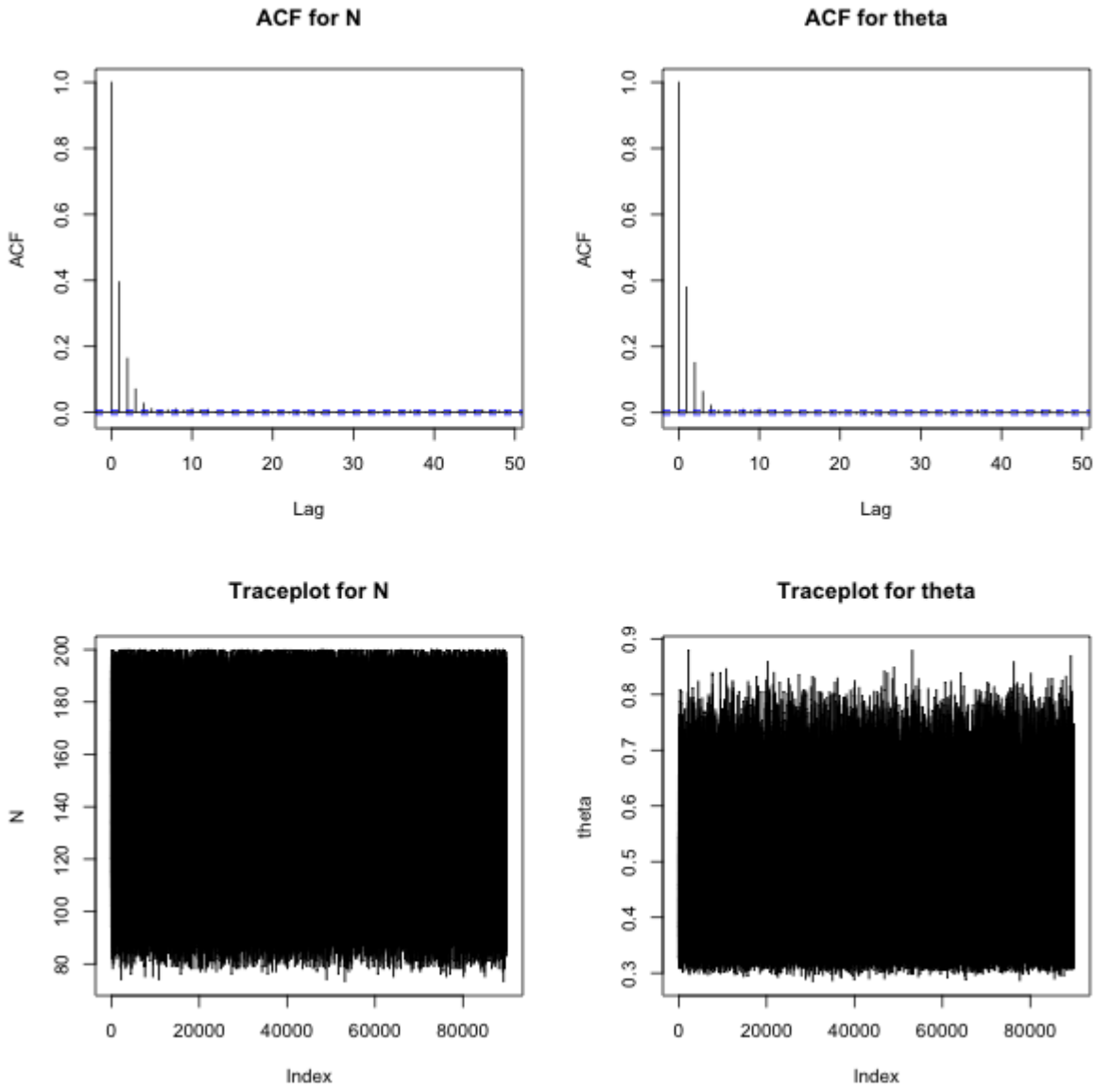
```
mcmc.mh <- function(y, mcmc.niters=1e4, geom.prob, norm.sd)
{
  # Complete with MH.
  S = sum(y)
  n = length(y)
  mcmc.chain <- matrix(, nrow=mcmc.niters, ncol=2)
  mcmc.chain[1,]=c(max(y),0.5)
  nacc <- 0
  N.bound = 500
  for(i in 2:mcmc.niters)
  {
    # 1. Current state
    N.old = mcmc.chain[i-1, 1]
    theta.old = mcmc.chain[i-1, 2]
    # 2. Propose new state
    #Sample new N
    repeat
    {
      #geometric over entire thing
        N.new = rgeom(1,1/(1+N.old))
      #only keep sample if N is in the boundaries
      if(N.new >= max(y) & N.new<=N.bound)
        break
    }
    #Sample new NTheta, then infer new theta
    repeat
    {
      #sample theta from Beta
      M.new = N.old * rbeta(1, shape1 = S+1, shape2 = n*N.old-S+1)
      theta.new = M.new/N.new
      #only keep theta if it is valid
      if(theta.new>=0 && theta.new<=1)
        break
    }

    # 3. Ratio
    mh.ratio = min(0, log.posterior(y,N.new,theta.new) -
                      log.posterior(y,N.old,theta.old))
    if(runif(1) < exp(mh.ratio)) {
      # Accept
      mcmc.chain[i, ] <- c(N.new, theta.new)
      nacc <- nacc + 1
    } else {
      mcmc.chain[i, ] <- c(N.old, theta.old)
    }
  }
  # Cut the burnin period.
  print(sprintf("Acceptance ratio %.2f%%", 100 * nacc / mcmc.niters))
  plot.chain(mcmc.chain)
  return(list("chain"=mcmc.chain, "accept" = 100 * nacc / mcmc.niters))
}
```



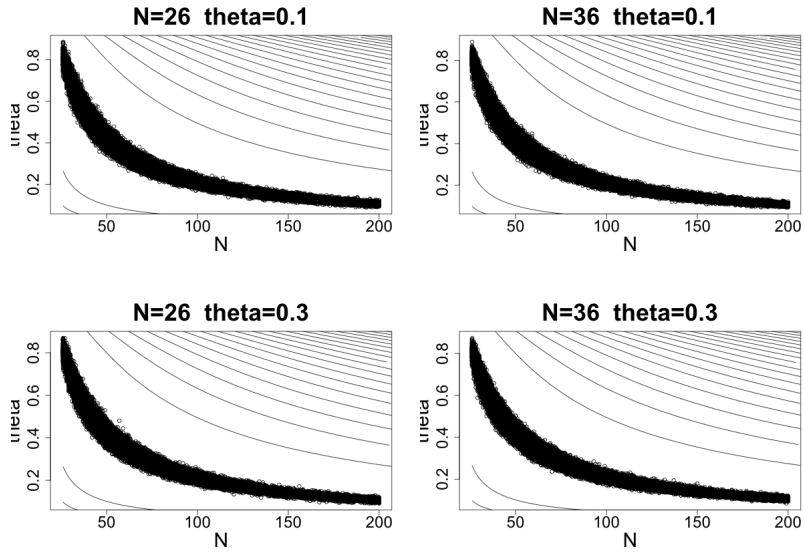


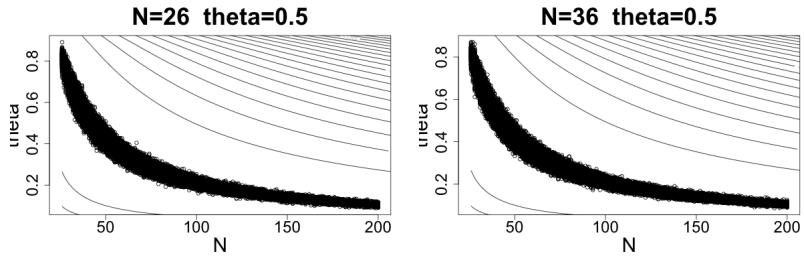
We note that the plots all seem to be very similar, so the starting point is irrelevant to the eventual sample. We include extra diagnostics plot for the last case ($N = 82$, $\theta = 0.9$) only because there would be too many plots otherwise. We inspect the autocorrelation functions of N and θ as well as their trace plots.

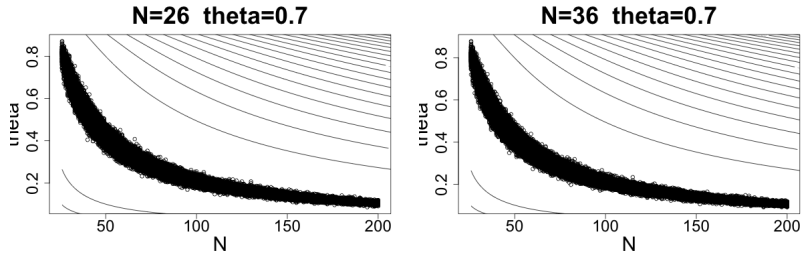


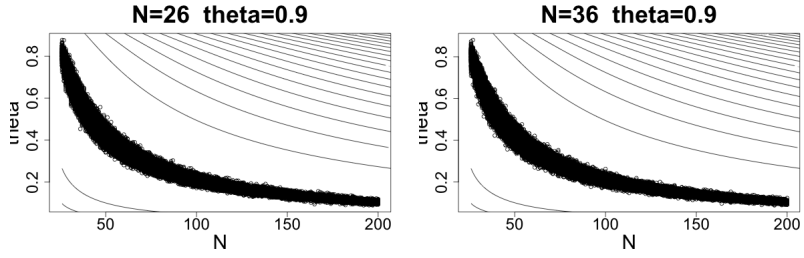
We see the ACF plots go to 0 pretty quickly, indicating that there is good mixing. The traceplots cover most of the range of the data, so this is a good sampler.

If we do the same thing with impala, we have

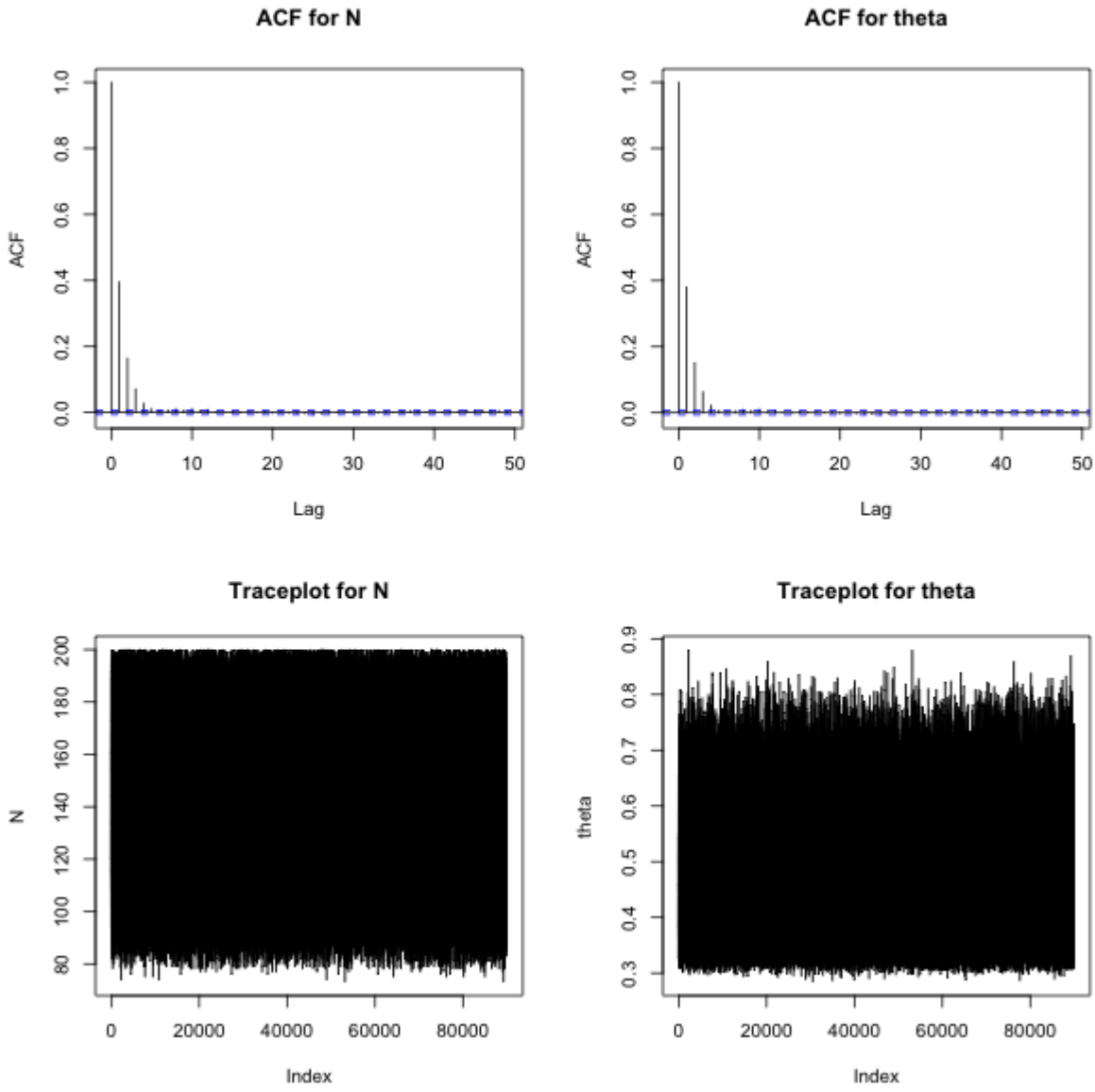








As before, we have that the plots are all very similar, so the starting point for the Metropolis-Hastings algorithm does not have much of an effect. For diagnostics we look at the ACF and trace plots for the last case:



We see the same thing as above, so this is a good sampler.

1.5 Recall that our posterior density of (N, θ) was

$$\mathbb{P}(N,\theta) \propto \left\{\prod_{i=1}^{n}\binom{N}{y_i}\right\}\frac{1}{N}\theta^S(1-\theta)^{nN-S}\mathbb{1}_{\theta\in[0,1]}$$

where $S = \sum_{i=1}^{n} y_i$. To find the marginal posterior distribution for N, we simply integrate out θ.

$$\begin{aligned}
\mathbb{P}(N|Y_1,\ldots,Y_n) &= \int_0^1 \mathbb{P}(N,\theta|Y_1,\ldots,Y_n)\ \mathrm{d}\theta \\
&\propto \int_0^1 \theta^S(1-\theta)^{nN-S}\left\{\prod_{i=1}^{n}\binom{N}{y_i}\right\}\frac{1}{N}\,\mathrm{d}\theta \\
&\propto \frac{\Gamma(S+1)\Gamma(nN-S+1)}{\Gamma(nN+2)}\left\{\prod_{i=1}^{n}\binom{N}{y_i}\right\}\frac{1}{N} \\
&\propto \frac{S!(nN-s)!}{(nN+1)!}\left\{\prod_{i=1}^{n}\binom{N}{y_i}\right\}\frac{1}{N} \\
&\propto \frac{1}{\binom{nN}{S}}\frac{1}{nN+1}\left\{\prod_{i=1}^{n}\binom{N}{y_i}\right\}\frac{1}{N}
\end{aligned}$$

Using this, we sum up the scaled marginal posterior for $N = \max(y), \max(y)+1, \ldots, \max(y)+200$. The number 200 is arbitrary here. We just want to have enough points so that the density becomes small enough. Dividing 1 by this sum gives us the normalizing constant.

```
impala = read.table("impala.txt", header = TRUE)
impala = impala[,1]

waterbuck = read.table("waterbuck.txt", header = TRUE)
waterbuck = waterbuck[,1]

unscaled.marg.post.N = function(N,y)
{
  S = sum(y)
  n = length(y)
  1/choose(n*N,S) * 1/(n*N+1) * prod(choose(N,y)/N)
}

y = waterbuck
N.seq = seq(max(y),max(y)+200,1)
marg.post.seq = sapply(N.seq, function(temp) unscaled.marg.post.N(temp,y))
waterbuck.constant = 1/sum(marg.post.seq)

y = impala
N.seq = seq(max(y),max(y)+1000,1)
marg.post.seq = sapply(N.seq, function(temp) unscaled.marg.post.N(temp,y))
impala.constant = 1/sum(marg.post.seq)

> waterbuck.constant
[1] 1.603238e+17
> impala.constant
[1] 2.955879e+13
```

So our normalizing constant for waterbuck is 1.6×10^{17} and for the impala is 2.96×10^{13}.

1.6 Now that we have our normalizing constants, we can renormalizing our marginal posterior densities for N. Once we do that, we can simply sum up all the values of the marginal posterior density where $N > 100$.

```
marg.post.N.waterbuck = function(N,y)
{
  S = sum(y)
  n = length(y)
  val = 1/choose(n*N,S) * 1/(n*N+1) * prod(choose(N,y)/N)*waterbuck.constant
  if(is.na(val))
    return(0)
  val
}
marg.post.N.impala = function(N,y)
{
  S = sum(y)
  n = length(y)
  val = 1/choose(n*N,S) * 1/(n*N+1) * prod(choose(N,y)/N)*impala.constant
  if(is.na(val))
    return(0)
  val
}

y = waterbuck
p.waterbuck = sum(sapply(101:7000,function(i) marg.post.N.waterbuck(i,y)))
p.waterbuck

y = impala
p.impala = sum(sapply(101:7000, function(i) marg.post.N.impala(i,y)))
p.impala
```

This gives

```
> p.waterbuck
[1] 0.6723058
> p.impala
[1] 0.003140613
```

So that for waterbuck, we have $\mathbb{P}(N > 100|Y_1, \ldots, Y_n) = 0.6723$ and for the impala, we have $\mathbb{P}(N > 100|Y_1, \ldots, Y_n) = 0.00314$. From our 10 posterior probability estimates, we have

```
> mean(waterbuck.prob)
[1] 0.8920223
> sd(waterbuck.prob)
[1] 0.001497997
> mean(impala.prob)
[1] 0.1521528
> sd(impala.prob)
[1] 0.002352665
```

Based on this, it looks like our analytical posterior probabilities do not match up with the simulated ones.