Empirical Bayes Analysis using MCMC

Greg Johnson

Consider the data $\mathbf{y} = (1.8, 1.6, 1.4, 1.6, 1.4, 1.5, 1.2)$.

We decide to model them as i.i.d. realizations of a gamma distribution:

$$y_i|\alpha, \beta \sim \text{Gamma}(\alpha, \beta)$$
$$f(y_i|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} y_i^{\alpha-1} \exp(-\beta y_i)$$

Suppose we want to perform an Empirical Bayes analysis where we model the parameters α and β as themselves random variables with prior distributions parameterized by known hyperparameters:

$$\alpha \sim \text{Gamma}(\alpha_\alpha = 2, \beta_\alpha = 0.5)$$

$$\beta \sim \text{Gamma}(\alpha_\beta = 1, \beta_\beta = 0.8)$$

Note that β is conjugate while α is not.

Given our prior distribution (encapsulating our prior knowledge of α and β) and our likelihood (summarizing the information about α and β coming from our data) we want to know the posterior density of α and β.

a)MCMC. Perform MCMC using a Gibbs step for β and a Metropolis Hastings step for α. What proposal distribution did you use?

Derive Joint Posterior

Using Bayes' theorem:

$$p(\alpha, \beta|\mathbf{y}) \propto p(\alpha, \beta) p(\mathbf{y}|\alpha, \beta)$$
$$\propto p(\alpha) p(\beta) \cdot l(\alpha, \beta)$$

Under our i.i.d. assumption, the likelihood is the product of 7 Gamma densities conditional on α and β:

$$l(\alpha, \beta) = f(\mathbf{y}|\alpha, \beta) = \prod_{i=1}^{7} f(y_i|\alpha, \beta) = \prod_{i=1}^{7} \text{Gamma}(\alpha, \beta)$$
$$= \prod_{i=1}^{7} \frac{\beta^\alpha}{\Gamma(\alpha)} y_i^{\alpha-1} \exp(-\beta y_i) = \frac{\beta^{n\alpha}}{[\Gamma(\alpha)]^n} \Big[\prod_{i=1}^{7} y_i\Big]^{\alpha-1} \exp\Big(-\beta \sum_{i=1}^{7} y_i\Big)$$

Our joint prior is the product of independent marginal Gamma priors for α and β:

$$p(\alpha, \beta) = \frac{(0.5)^2}{\Gamma(2)} \alpha^{2-1} \exp(-0.5\alpha) \times \frac{(0.8)^1}{\Gamma(1)} \beta^{1-1} \exp(-0.8\beta)$$

Now we may turn our attention back to the posterior:

$$p(\alpha,\beta|\mathbf{y}) \propto \frac{(0.5)^2}{\Gamma(2)}\alpha^{2-1}\exp\big(-0.5\alpha\big) \times \frac{(0.8)^1}{\Gamma(1)}\beta^{1-1}\exp\big(-0.8\beta\big) \times \frac{\beta^{n\alpha}}{[\Gamma(\alpha)]^n}\Big[\prod_{i=1}^{7} y_i\Big]^{\alpha-1}\exp\Big(-\beta\sum_{i=1}^{7} y_i\Big)$$

$$\propto \Big[\beta^{n\alpha}\exp\Big(-\big(\sum y_i + 0.8\big)\beta\Big)\Big]\Big[\alpha\big[\Gamma(\alpha)\big]^{-n}\big(\prod y_i\big)^{\alpha-1}\exp\Big(-0.5\alpha\Big)\Big]$$

If we recognize the gamma kernel in the left term, we can factor the posterior into a marginal and conditional! Specifically we recognize $\text{Gamma}\Big(n\alpha+1, \sum y_i + 0.8\Big)$. Introducing the constant let's us complete the factorization.

$$p(\alpha,\beta|\mathbf{y}) \propto \Big[\frac{(\sum y_i + 0.8)^{n\alpha+1}}{\Gamma(n\alpha+1)}\beta^{n\alpha}\exp\Big(-\big(\sum y_i + 0.8\big)\beta\Big)\Big]\Big[\frac{\Gamma(n\alpha+1)}{(\sum y_i + 0.8)^{n\alpha+1}}\alpha\big[\Gamma(\alpha)\big]^{-n}\big(\prod y_i\big)^{\alpha-1}\exp\Big(-0.5\alpha\Big)\Big]$$

$$p(\alpha,\beta|\mathbf{y}) \propto p(\beta|\alpha,\mathbf{y})p(\alpha|\mathbf{y})$$

Outline MCMC Algorithm

Since $\beta|\alpha,\mathbf{y}$ follows a known parameterized family, we will use a Gibbs step on it, sampling from $\text{Gamma}\Big(n\alpha+1, \sum y_i+0.8\Big)$, conditional on the $\alpha|\mathbf{y}$ sampled using a Metropolis-Hastings step. Since $\alpha|\mathbf{y}$ follows a distribution not commonly sampled from, we are forced to use MH; fortunately the density is easy to evaluate.

Our MCMC algorithm will proceed as follows:

1. Initialize α^0 from an overdispersed distribution whose support covers likely values of α. Sample β^0 from $\text{Gamma}\Big(n\alpha+1, \sum y_i + 0.8\Big)$.
2. For $t = 1, 2, ...$, we first perform a Metropolis-Hastings step for α. We will sample a proposal α^* from the jumping distribution $J_t(\alpha^*|\alpha^{t-1})$, compute the ratio r and then update α^{t-1} to α^* with probability $\min(r, 1)$.
3. If α^t is updated, then we perform the Gibbs step, sampling: $\beta^t \sim \text{Gamma}(n\alpha^t + 1, \sum y_i + 0.8)$.

Select a Jumping Distribution

To perform Metropolis-Hastings, we need a proposal distribution. When the posterior support is the real line, the Gaussian is usually a good place to start. In this case, α is a Gamma parameter and lives only on the positive reals; to match supports, we can use the log-normal distribution:

$$X \sim \text{N}(\mu, \sigma^2)$$
$$Y = h(X) = \exp(X) \sim \text{LogN}(\mu, \sigma^2)$$
$$h : \mathbb{R} \to \mathbb{R}^+$$
$$f_Y(y) = y^{-1}f_X(\log(y)) = (2\pi\sigma^2)^{-1/2}y^{-1}\exp\Big\{-\frac{1}{2\sigma^2}\Big(\log(y)-\mu\Big)^2\Big\}$$

Any good jumping distribution should jump to a value that is close to the θ^{t-1} that it is conditioned upon. When using the normal distribution, we can set $\mu = \theta^{t-1}$. For the log-normal distribution, the expectation doesn't equal the mean parameter (and it's entangled with the variance parameter) so to make sure our MH step is expected to jump in the right place, we need to be very careful with how we set the mean and variance of the jumping log-normal. The log-normal moments are:

$$\mathrm{E}(Y) = \exp\left\{\mu + \sigma^2/2\right\}$$
$$\mathrm{Var}(Y) = \Big(\exp\left\{\sigma^2\right\} - 1\Big)\exp\left\{2\mu + \sigma^2\right\} = \Big(\exp\left\{\sigma^2\right\} - 1\Big)\big[E(Y)\big]^2$$

Inverting to solve for μ and σ^2:

$$\mu = \log\big(E(Y)\big) - \frac{1}{2}\log\left(\frac{\mathrm{Var}(Y)}{(E(Y))^2} + 1\right)$$
$$\sigma^2 = \log\left(\frac{\mathrm{Var}(Y)}{(E(Y))^2} + 1\right)$$

We want $\mathrm{E}(Y)$ to be set to the previous iteration α^{t-1} and $\mathrm{Var}(Y)$ to be our acceptance tuning parameter ψ for our MCMC. That means when we sample a proposal from the log-mean distribution and evaluate densities of the log-mean distribution for the ratio r, we use the following transformations of α^{t-1} and ψ to set as the log-mean parameters:

$$\mu = \log\big(E(Y)\big) - \frac{1}{2}\log\left(\frac{\mathrm{Var}(Y)}{(E(Y))^2} + 1\right)$$
$$\sigma^2 = \log\left(\frac{\mathrm{Var}(Y)}{(E(Y))^2} + 1\right)$$
$$\mathrm{LogN}\left\{\mu(\alpha^{t-1},\psi) = \log(\alpha^{t-1}) - \frac{1}{2}\log\left(\frac{\psi}{(\alpha^{t-1})^2} + 1\right), \sigma^2(\alpha^{t-1},\psi) = \log\left(\frac{\psi}{(\alpha^{t-1})^2} + 1\right)\right\}$$

And our ratio will take the shape:

$$r = \frac{p(\alpha^*|\mathbf{y})}{p(\alpha^{t-1}|\mathbf{y})} \cdot \frac{\mathrm{LogN}\big(\mu(\alpha^*,\psi),\sigma^2(\alpha^*,\psi)\big)}{\mathrm{LogN}\big(\mu(\alpha^{t-1},\psi),\sigma^2(\alpha^{t-1},\psi)\big)}$$

Implement MCMC

```
data = c(1.8, 1.6, 1.4, 1.6, 1.4, 1.5, 1.2)

# define density functions for mcmc

# let's look at the unnormalized alpha posterior and the lognormal jumping
# distribution xplot = seq(.1,20,length.out=100) yplot = alphapost(xplot)
# yplot = yplot/sum(yplot)
# plot(xplot,yplot,type='l',ylab='density',xlab=expression(alpha))
# curve(dlnorm(x,meanlog=4,sdlog=4),add=TRUE,col='red',lwd=2)



mcmc = function(start, ndraws, y, varjump) {
    # cache functions of data
    sumy = sum(y)
    prody = prod(y)
    n = length(y)
```

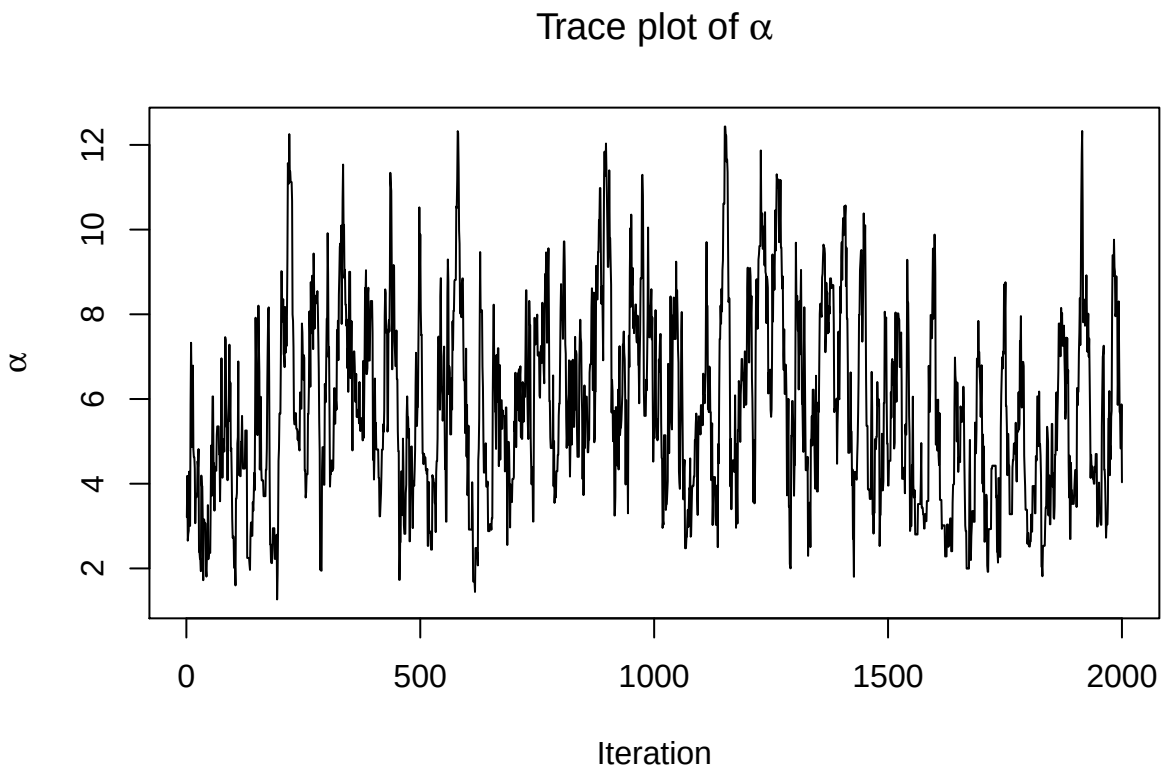
```
    # unnormalized posterior marginal density of alpha avoid overflow
    alphapost = function(alpha) {
        exp(log(gamma(n * alpha + 1)) - log((sumy + 0.8)^(n * alpha + 1)) +
            log(alpha) + log((gamma(alpha))^(-n)) + log(prody^(alpha - 1)) +
            log(exp(-0.5 * alpha)))
    }

    # jumping distribution
    rjump = function(mean) {
        meanlog = log(mean) - 0.5 * log(varjump/mean^2 + 1)
        sdlog = sqrt(log(varjump/(mean)^2 + 1))
        rlnorm(1, meanlog = meanlog, sdlog = sdlog)
    }
    djump = function(x, mean) {
        meanlog = log(mean) - 0.5 * log(varjump/mean^2 + 1)
        sdlog = sqrt(log(varjump/(mean)^2 + 1))
        dlnorm(x, meanlog = meanlog, sdlog = sdlog)
    }


    # initialize Markov chain
    alpha = start

    # save draws and acceptance indicator
    post_draws = matrix(NA, ndraws, 2)
    colnames(post_draws) = c("alpha", "beta")
    acceptance = rep(0, ndraws)

    for (j in 1:ndraws) {

        # perform MH step to sample alpha
        proposal = rjump(alpha)
        r = exp(log(alphapost(proposal)) - log(alphapost(alpha)) + log(djump(alpha,
            proposal)) - log(djump(proposal, alpha)))

        if (runif(1) < min(r, 1)) {
            alpha = proposal  #update alpha
            acceptance[j] = 1
        }

        # perform Gibbs step to sample beta|alpha
        beta = rgamma(1, shape = n * alpha + 1, rate = sumy + 0.8)

        # save draw
        post_draws[j, ] = c(alpha, beta)
    }

    return(list(post_draws = post_draws, aratio = mean(acceptance)))
}

# test run starting point - alpha prior mean = 2/.5 = 4
output = mcmc(start = 4, ndraws = 2000, y = data, varjump = 2)
output$aratio
```

```
## [1] 0.7455
```

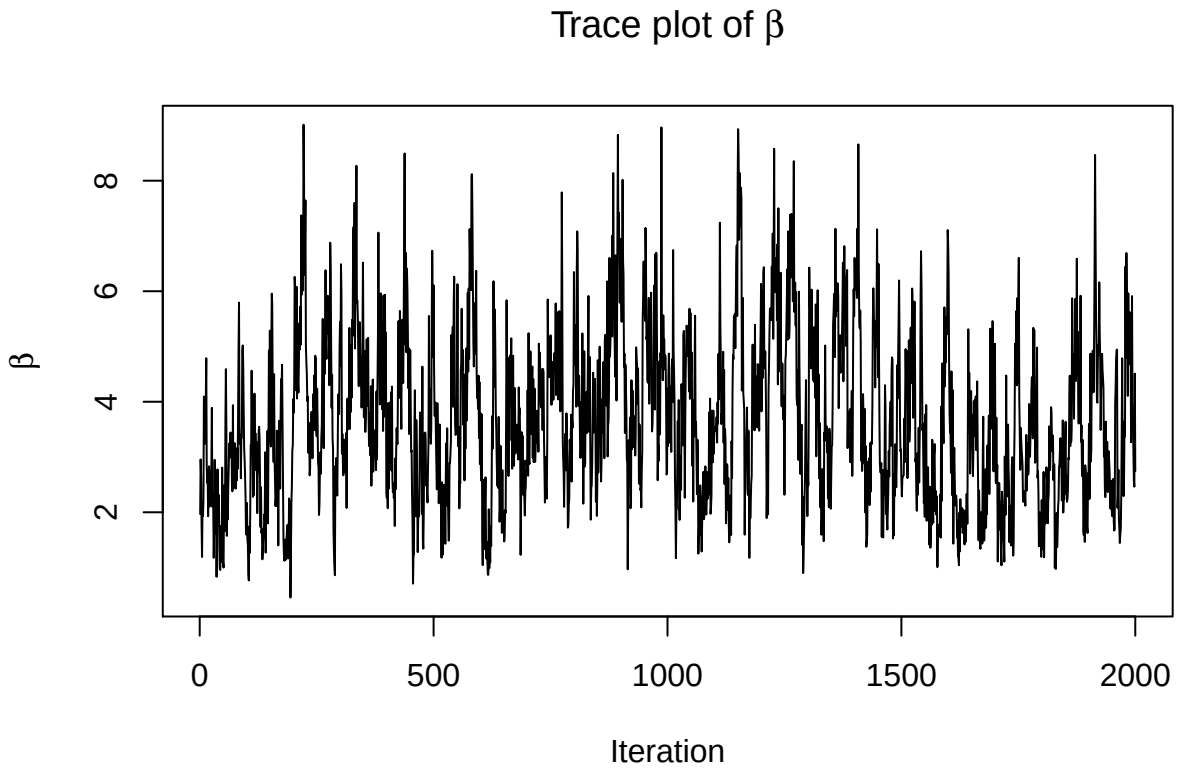
b) Convergence. Provide trace plots and autocorrelation plots of joint posterior draws. Discuss. Report acceptance ratio, burn-in, and thinning/effective sample size. Tune the MCMC to maximize acceptance while achieving good mixing (low autocorrelation).

Diagnostic Plots

```
plot(output$post_draws[, "alpha"], type = "l", xlab = "Iteration", ylab = expression(alpha),
    main = expression(paste("Trace plot of ", alpha)))
```

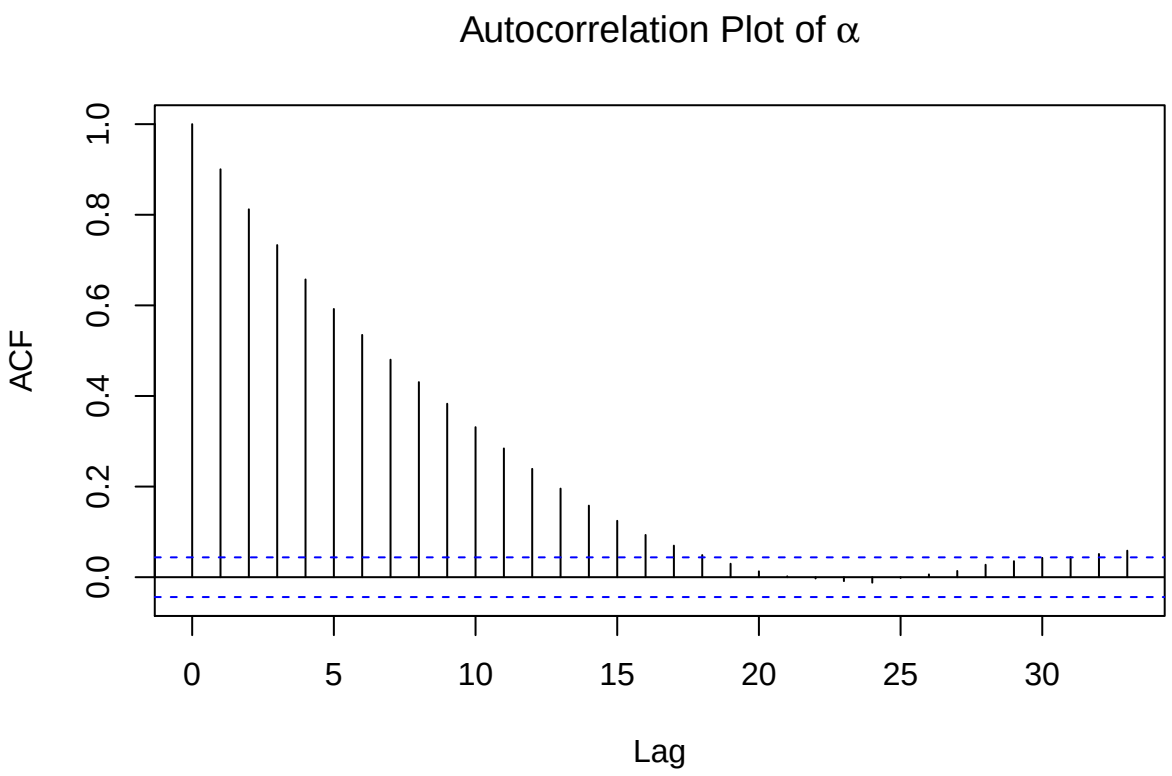



```
plot(output$post_draws[, "beta"], type = "l", xlab = "Iteration", ylab = expression(beta),
    main = expression(paste("Trace plot of ", beta)))
```

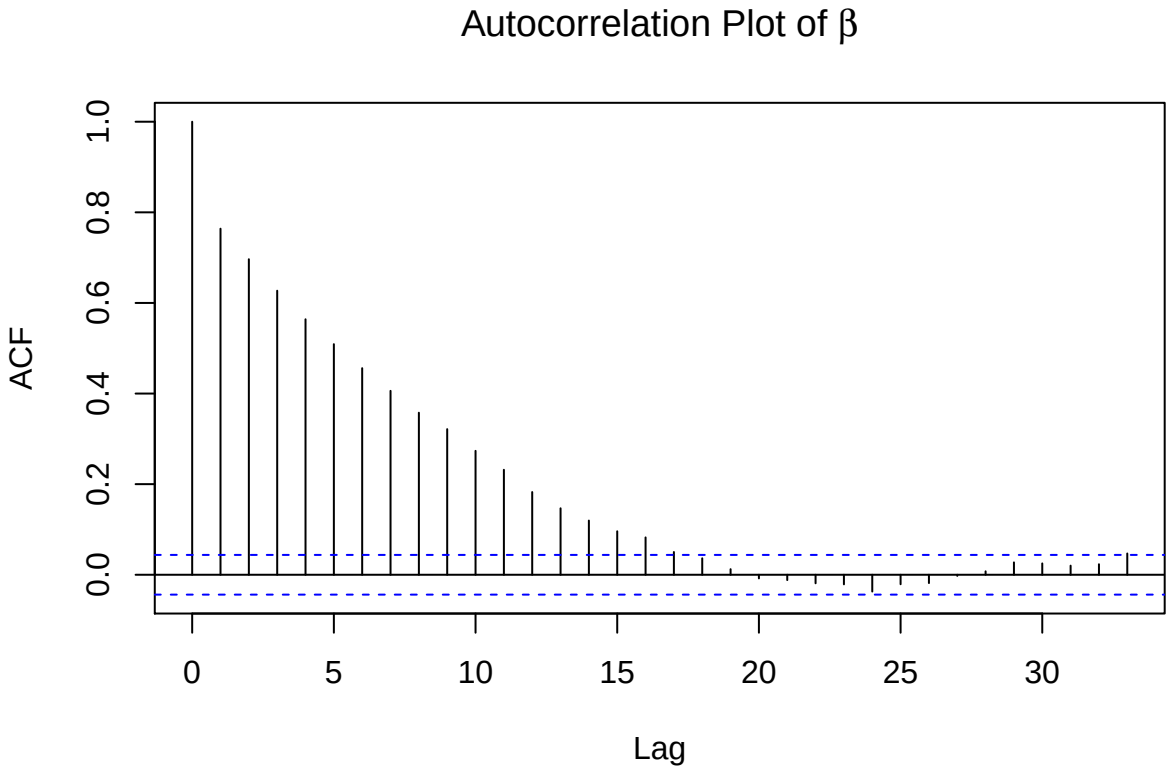


It looks like stationarity is achieved pretty immediately. This is most likely because our starting α was a point in the posterior density with high probability. I don't believe we need any burn-in.

```
acf(output$post_draws[, "alpha"], main = expression(paste("Autocorrelation Plot of ",
    alpha)))
```



```
acf(output$post_draws[, "beta"], main = expression(paste("Autocorrelation Plot of ",
    beta)))
```



Univariate autocorrelation is significant up to *about* the 22nd lag for both α and β.

Acceptance Ratio, Burn-in, and Thinning

For $\psi = \text{Var}(Y) = 2$ for our log-normal jumping distribution, we achieve an acceptance ratio of 0.7455 which is on the high side. As noted below the trace plots, with our starting α, it doesn't look like we need to perform any burn-in. Autocorrelation is a problem - however thinning down to every 20th draw completely removes any autocorrelation effect. Unfortunately this gives us a very small effective sample size (<100). It appears that our MCMC isn't jumping enough between draws - hence the significant autocorrelations for the 22nd lag! Our next step should be to tune the variance parameter of our jumping distribution to find the optimal tradeoff between autocorrelation and acceptance ratio/effective sample size.

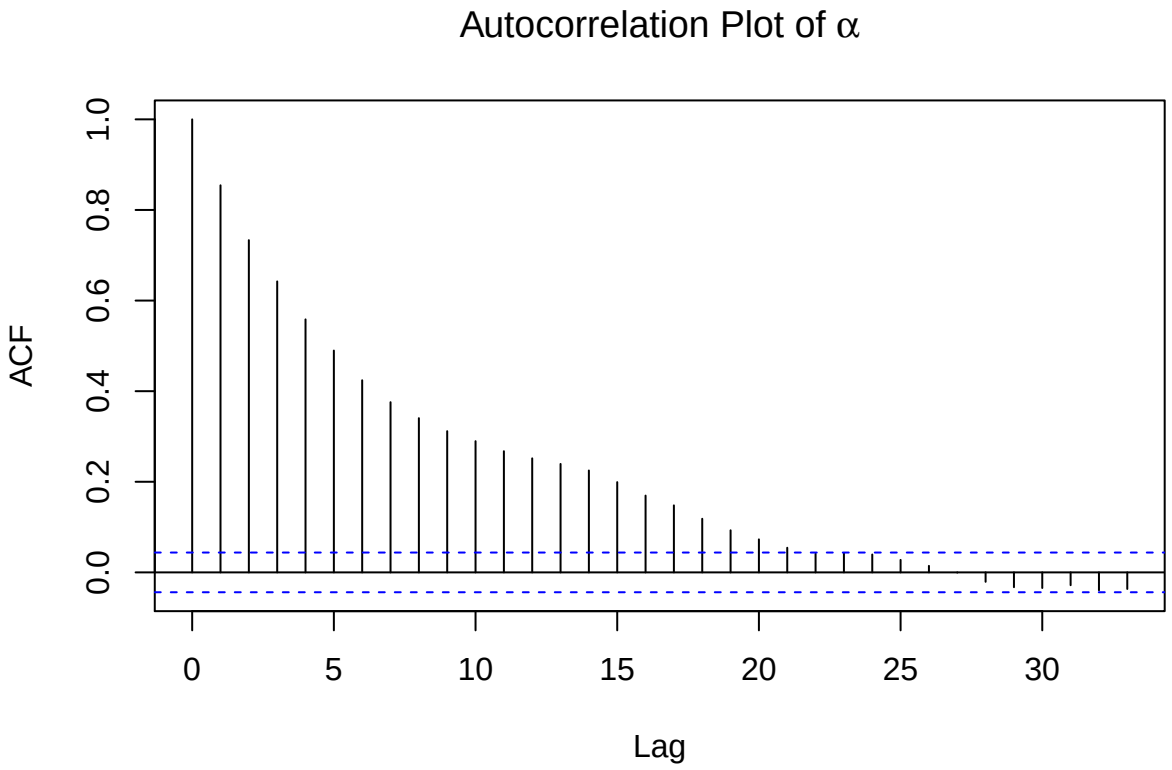
Tune MCMC

We need to explore our MCMC's behavior over a grid of values of $\psi = \text{Var}(Y)$. Based on the slow mixing of $\psi = 2$, we should start at 2 and increase.

```
# Varjump 4
output.4 = mcmc(start = 4, ndraws = 2000, y = data, varjump = 4)
output.4$aratio
```

```
## [1] 0.6675
```

```
acf(output.4$post_draws[, "alpha"], main = expression(paste("Autocorrelation Plot of ",
    alpha)))
```




```
# Varjump 6
set.seed(31)
output.6 = mcmc(start = 4, ndraws = 2000, y = data, varjump = 6)
output.6$aratio
```

```
## [1] 0.605
```

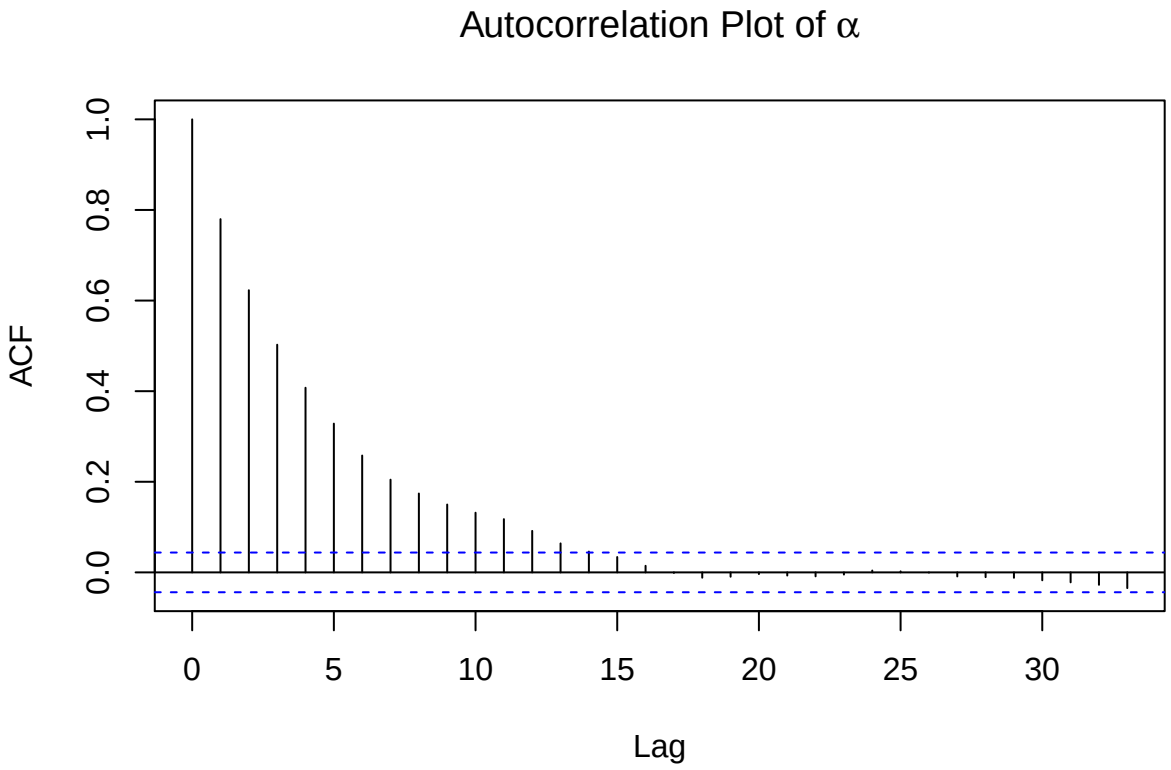
```
# Varjump 8
set.seed(31)
output.8 = mcmc(start = 4, ndraws = 2000, y = data, varjump = 8)
output.8$aratio
```

```
## [1] 0.5715
```

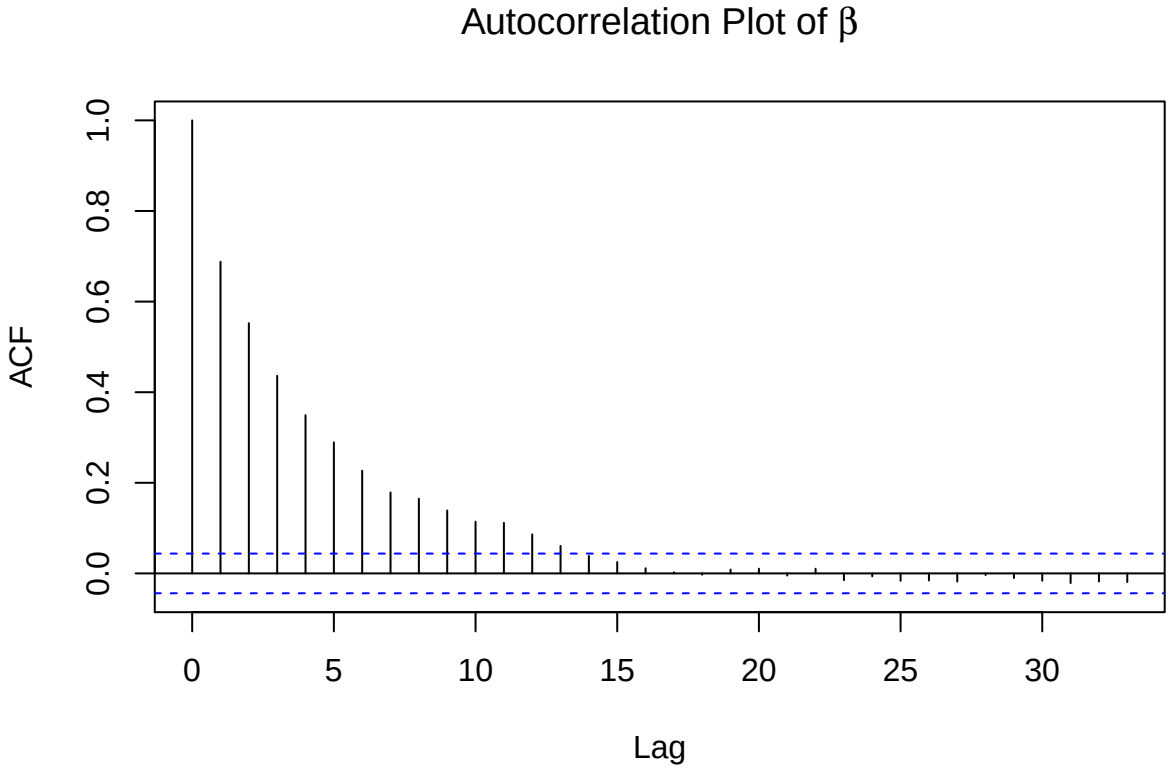
```
set.seed(31)
output.10 = mcmc(start = 4, ndraws = 2000, y = data, varjump = 10)
output.10$aratio
```

```
## [1] 0.543
```

```
acf(output.10$post_draws[, "alpha"], main = expression(paste("Autocorrelation Plot of ",
    alpha)))
```




```
acf(output.10$post_draws[, "beta"], main = expression(paste("Autocorrelation Plot of ",
    beta)))
```



Unfortunately the highest value of ψ that I could test was 10. Any higher and the jumping distribution started to jump to large values of α that created insurmountable overflow issues when the $\Gamma(n\alpha + 1)$ term

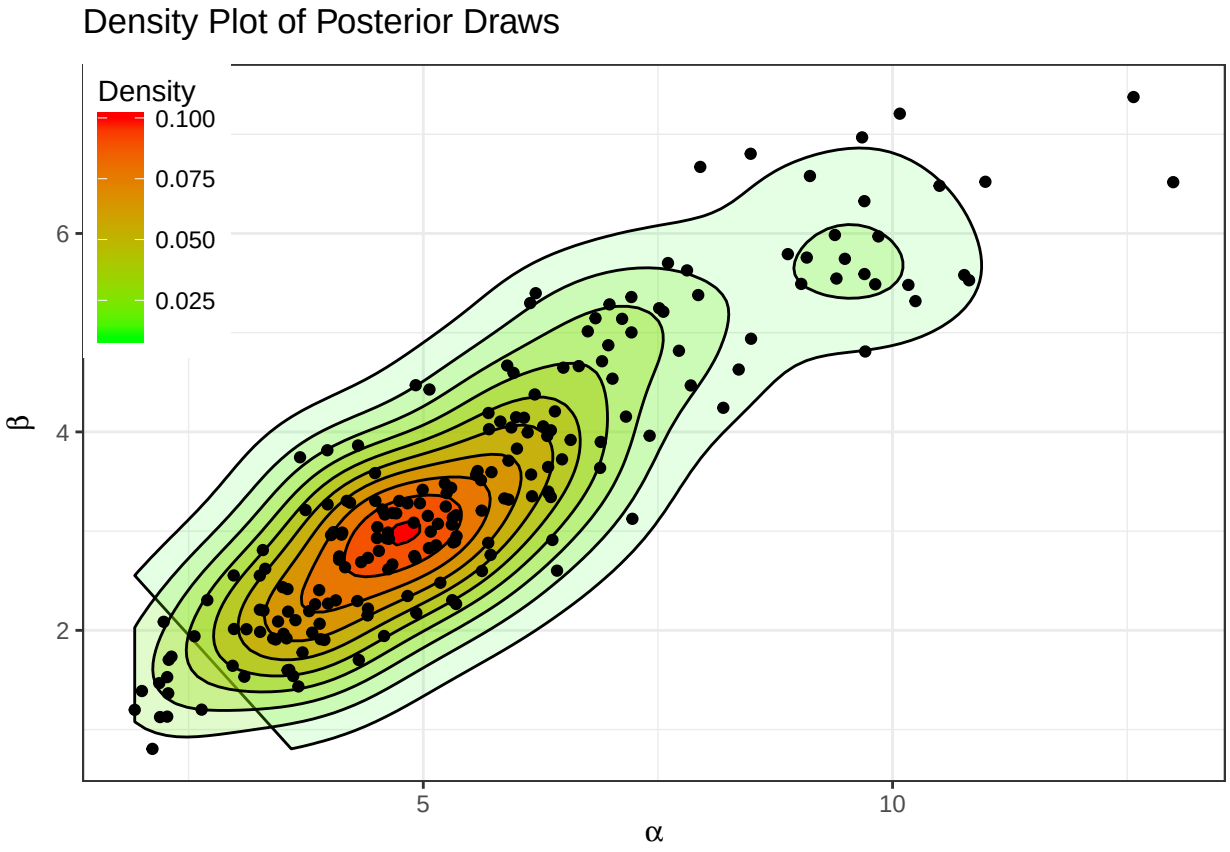
was evaluated in the unnormalized posterior density $p(\alpha|\mathbf{y})$. However the trend in decreasing acceptance ratio can be seen as the variance of the jumping distribution is increased. At $\psi = 10$ the autocorrelation drag has dropped to about 10. This greatly decreases the magnitude of required thinning and therefore increases the effective sample size.

c) Posterior Uncertainty. Obtain posterior plots for 200 draws. Also report the MAP and 95% credible intervals for model parameters.

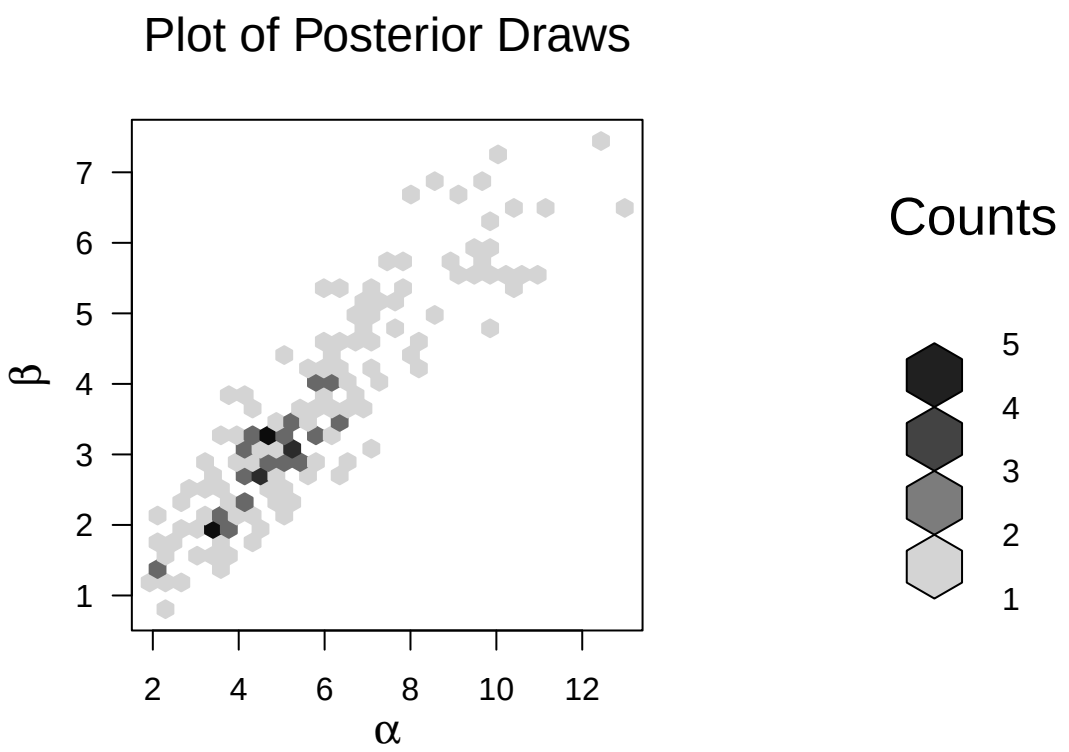
```
set.seed(31)
output = mcmc(start=4,ndraws=2000,y=data,varjump=10)
post_draws = output$post_draws[seq(1,2000,10),]
```

Posterior Plots

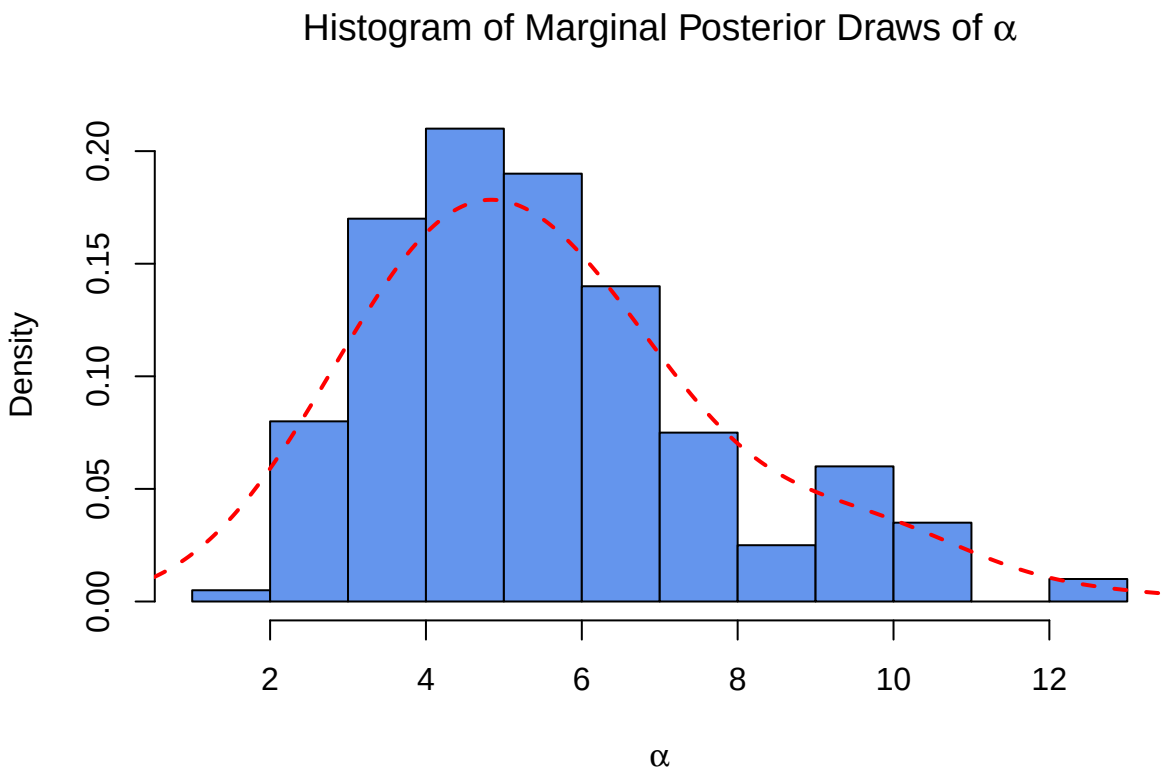
```
commonTheme = list(labs(color = "Density", fill = "Density", x = expression(alpha),
    y = expression(beta)), theme_bw(), theme(legend.position = c(0, 1), legend.justification = c(0,
    1)))
ggplot(data = as.data.frame(post_draws), aes(alpha, beta)) + stat_density2d(aes(fill = ..level..,
    alpha = ..level..), geom = "polygon", colour = "black") + scale_fill_continuous(low = "green",
    high = "red") + guides(alpha = "none") + geom_point() + commonTheme + ggtitle(expression(paste("Dens
```




```
plot(hexbin(post_draws[, "alpha"], post_draws[, "beta"]), xlab = expression(alpha),
    ylab = expression(beta), main = "Plot of Posterior Draws")
```

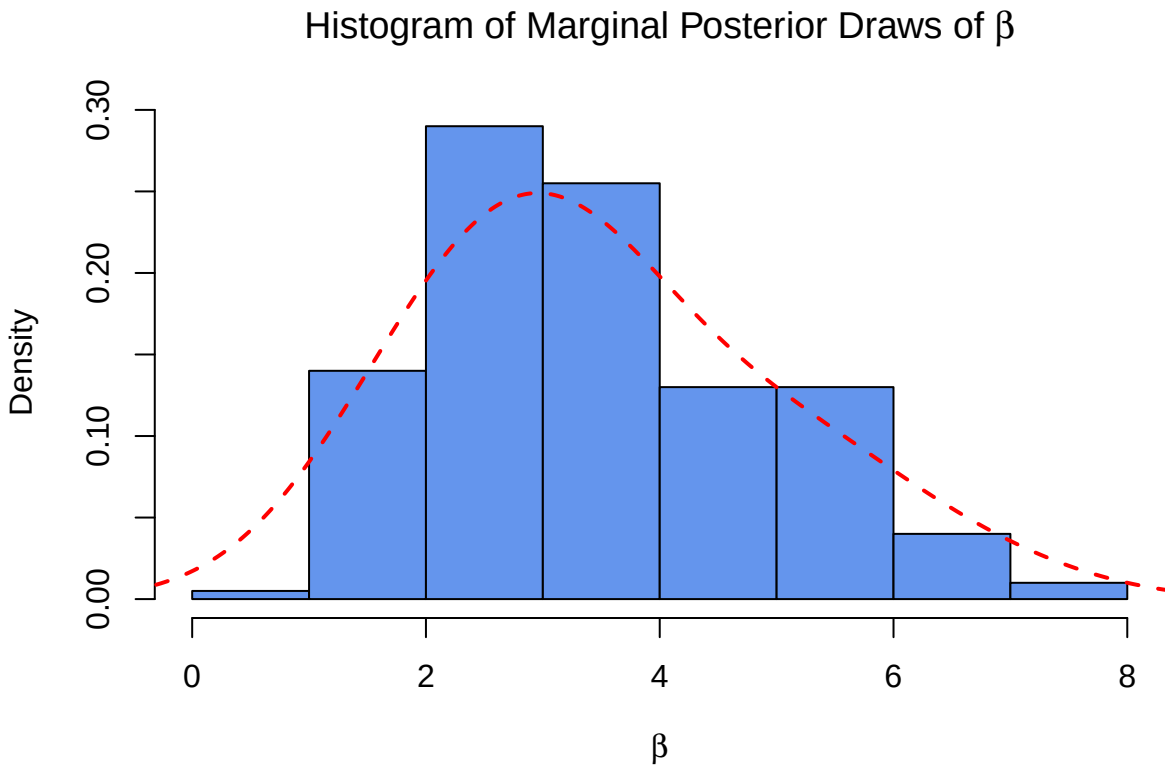



```
hist(post_draws[, "alpha"], main = expression(paste("Histogram of Marginal Posterior Draws of ",
    alpha)), xlab = expression(alpha), freq = FALSE, col = "cornflowerblue")
lines(density(post_draws[, "alpha"], adjust = 2), col = "red", lwd = 2, lty = 2)
```




```
hist(post_draws[, "beta"], main = expression(paste("Histogram of Marginal Posterior Draws of ",
    beta)), xlab = expression(beta), freq = FALSE, col = "cornflowerblue")
```

```
lines(density(post_draws[, "beta"], adjust = 2), col = "red", lwd = 2, lty = 2)
```



MAP & 95% Credible Intervals

```
#alpha
alpha_CI = quantile(post_draws[,"alpha"],probs=c(0.025,0.975))
alpha_CI
```

```
##      2.5%     97.5%
##  2.233555 10.506985
```

```
alpha_MAP = post_draws[,"alpha"][which.max(density(post_draws[,"alpha"])$y)]
alpha_MAP
```

```
## [1] 5.296703
```

```
#beta
beta_CI = quantile(post_draws[,"beta"],probs=c(0.025,0.975))
beta_CI
```

```
##     2.5%    97.5%
## 1.361525 6.581434
```

```
beta_MAP = post_draws[,"beta"][which.max(density(post_draws[,"beta"])$y)]
beta_MAP
```

```
## [1] 1.701259
```

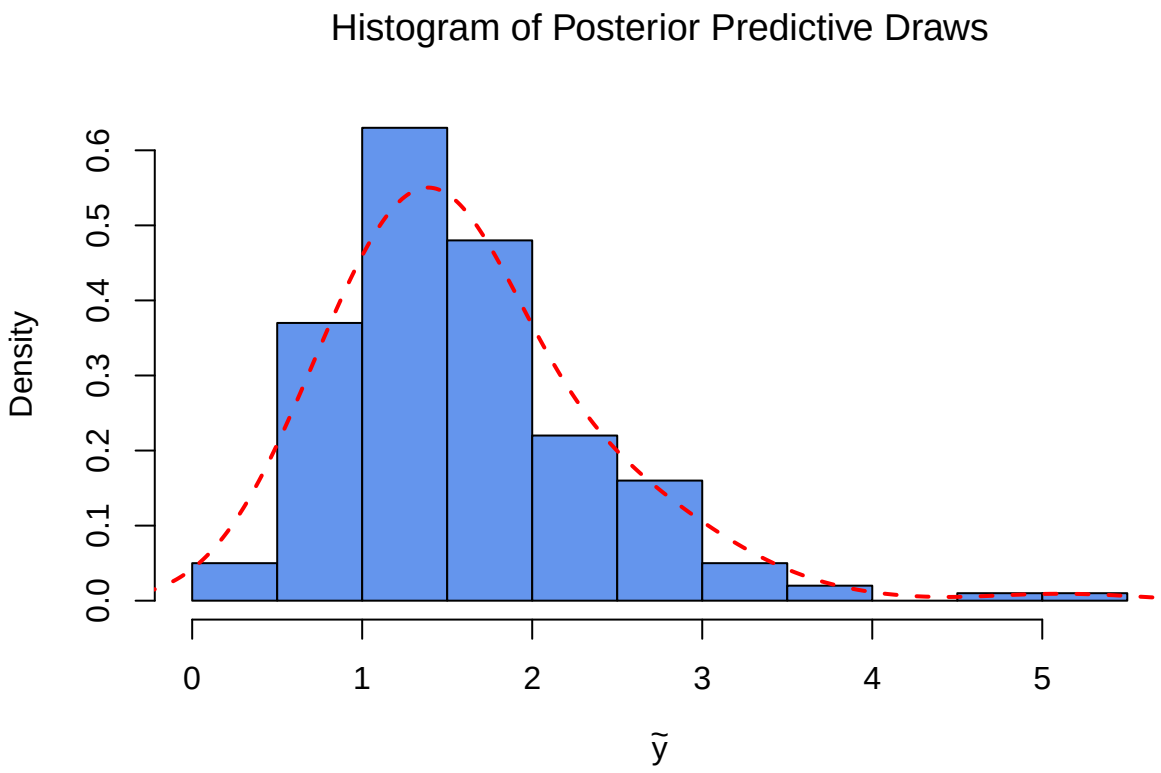
Given our data and our prior beliefs about α and β, there is a 95% probability that α is between 2.23 and 10.51. The mode of the marginal posterior for α (the data point whose neighborhood is most likely to contain the true parameter value) is 5.3. There is a 95% probability that β is between 1.36 and 6.58. The mode of the marginal posterior for β is 5.3.

d) Posterior Predictive. Use posterior samples to simulate the posterior predictive distribution. Provide the density plot and report the MAP and 95% predictive credible interval.

```
post_pred = numeric()
for (j in 1:nrow(post_draws)) {
    post_pred[j] = rgamma(1, shape = post_draws[j, "alpha"], rate = post_draws[j,
        "beta"])
}
```

Posterior Predictive Plot

```
hist(post_pred, main = expression(paste("Histogram of Posterior Predictive Draws")),
    xlab = expression(tilde(y)), freq = FALSE, col = "cornflowerblue")
lines(density(post_pred, adjust = 2), col = "red", lwd = 2, lty = 2)
```



MAP & 95% Predictive Credible Intervals

```
post_pred_CI = quantile(post_pred,probs=c(0.025,0.975))
post_pred_CI
```

```
##      2.5%     97.5%
## 0.5096546 3.2573914
```

```
post_pred_MAP = post_pred[which.max(density(post_pred)$y)]
post_pred_MAP
```

```
## [1] 2.533445
```

We expect future observations to have a 95% chance of falling within the interval 0.51 and 3.26. The MAP estimate is 2.53.