Q1 Simplified Loglikelihood

```
diff <- rep(0,10)
n <- 100
y <- runif(n,0,1)
logy <- log(y)
for (i in 1:10) {
  alpha <- runif(1,0,1)
  beta <- runif(1,0,1)
  lambda <- runif(1,0,1)
  eta <- beta^lambda
  r1 <- sum(dggamma(y,alpha,beta,lambda,log=TRUE))
  r2 <- ggamma.loglik(alpha,eta,lambda,logy)
  diff[i] <- r1-r2
}

diff
```

```
##  [1] 111.0751 111.0751 111.0751 111.0751 111.0751 111.0751 111.0751
##  [8] 111.0751 111.0751 111.0751
```

As we can see, for fixed $\boldsymbol{Y}$, no matter what the values of alpha, beta, and lambda are, the difference between the simplified likelihood $l(\alpha, \eta, \lambda \mid \boldsymbol{Y}) = n \cdot \big[\log \lambda - \log \Gamma(\alpha\lambda^{-1}) + \alpha\lambda^{-1} \log \eta\big] + \alpha S - \eta T_\lambda$ and the unsimplified version $l(\alpha, \eta, \lambda \mid \boldsymbol{Y}) = \sum_{i=1}^n \log f(Y_i \mid \alpha, \eta, \lambda)$ is a constant.

Q2 Conditional Distribution is a Gamma Distribution

$$
\begin{aligned}
\mathcal{L}(\alpha, \eta, \lambda \mid \boldsymbol{Y}) &= \exp\{l(\alpha, \eta, \lambda \mid \boldsymbol{Y})\} \\
&= \exp\{log\lambda^n - \log \Gamma(\alpha\lambda^{-1})^n + \log \eta^{n\alpha\lambda^{-1}} + \alpha S - \eta T_\lambda\} \\
&= \frac{\lambda^n}{\Gamma(\alpha\lambda^{-1})^n} \cdot \eta^{n\alpha\lambda^{-1}} \cdot \exp\{\alpha\mathcal{S}\} \cdot \exp\{-\eta T_\lambda\}
\end{aligned}
$$

then,

$$
\mathcal{L}(\alpha, \eta, \lambda \mid \boldsymbol{Y}) \cdot g(\alpha, \lambda) = \frac{\lambda^n}{\Gamma(\alpha\lambda^{-1})^n} \cdot \eta^{n\alpha\lambda^{-1}} \cdot \exp\{\alpha\mathcal{S}\} \cdot \exp\{-\eta T_\lambda\} \cdot g(\alpha, \lambda)
$$

Throwing out everything that doesn't depend on η, we get

$$
\eta^{n\alpha\lambda^{-1}} \cdot \exp\{-\eta T_\lambda\} = \eta^{\hat{\kappa}-1} \cdot \exp\{-\hat{\gamma}\eta\}
$$

Therefore, $\hat{\kappa} = n\alpha\lambda^{-1} + 1$ and $\hat{\gamma} = T_\lambda = \sum_{i=1}^n Y_i^\lambda$

Q3 Marginal Posterior Distribution

The function `ggamma.logmarg` should return

$$
\begin{aligned}
&\log \frac{\mathcal{L}(\alpha,\eta,\lambda \mid \boldsymbol{Y})}{\texttt{dgamma}(\texttt{x}=\eta,\texttt{shape}=\hat{\kappa},\texttt{rate}=\hat{\gamma})}\\
=&l(\alpha,\eta,\lambda \mid \boldsymbol{Y}) - \log(\texttt{dgamma}(\texttt{x}=\eta,\texttt{shape}=\hat{\kappa},\texttt{rate}=\hat{\gamma}))\\
=&n\cdot\left[\log\lambda - \log\Gamma(\alpha\lambda^{-1}) + \alpha\lambda^{-1}\log\eta\right] + \alpha S - \eta T_\lambda - \log(\frac{\hat{\gamma}^{\hat{\kappa}}}{\Gamma(\hat{\kappa})}\cdot\eta^{\hat{\kappa}-1}\exp\{\eta\hat{\gamma}\})\\
=&n\log(\lambda) - n\log\Gamma(\alpha\lambda^{-1}) + n\alpha\lambda^{-1}\log\eta + \alpha S - \eta T_\lambda - \left[\log(\hat{\gamma}^{\hat{\kappa}}) - \log\Gamma(\hat{\kappa}) + \log(\eta^{\hat{\kappa}-1}) - \eta\hat{\gamma}\right]\\
=&n\log(\lambda) - n\log\Gamma(\alpha\lambda^{-1}) + n\alpha\lambda^{-1}\log\eta + \alpha S - \eta T_\lambda - \hat{\kappa}\log(\hat{\gamma}) + \log\Gamma(\hat{\kappa}) - (\hat{\kappa}-1)\log\eta + \eta\hat{\gamma}\\
&(\text{Since } \hat{\kappa} = n\alpha\lambda^{-1}+1 \text{ and } \hat{\gamma} = T_\lambda = \sum_{i=1}^{n} Y_i^{\lambda})\\
=&n\log(\lambda) - n\log\Gamma(\alpha\lambda^{-1}) + \alpha S - \hat{\kappa}\log(\hat{\gamma}) + \log\Gamma(\hat{\kappa})
\end{aligned}
$$

```r
ratio <- rep(0,10)
n <- 100
y <- runif(n,0,1)
logy <- log(y)

for (i in 1:10) {
  alpha <- runif(1,0,1)
  beta <- runif(1,0,1)
  lambda <- runif(1,0,1)
  eta <- beta^lambda
  khat <- n*alpha/lambda+1
  ghat <- sum(exp(lambda*logy))

  loglik <- ggamma.loglik(alpha,eta,lambda,logy)
  logmarg <- ggamma.logmarg(alpha,lambda,logy,khat,ghat)
  logcond <- dgamma(eta,shape=khat,rate=ghat,log = TRUE)

  ratio[i] <- loglik - (logmarg + logcond)
}

ratio
```

```
##  [1]  0.000000e+00  5.684342e-14 -5.684342e-14 -5.684342e-14  1.136868e-13
##  [6]  0.000000e+00  0.000000e+00 -4.547474e-13  2.842171e-14 -2.842171e-13
```

As showed above, for fixed $\boldsymbol{Y}$, no matter what the values of alpha, beta, and lambda are, the difference between the simplified likelihood $l(\alpha,\eta,\lambda \mid \boldsymbol{Y}) = n\cdot\left[\log\lambda - \log\Gamma(\alpha\lambda^{-1}) + \alpha\lambda^{-1}\log\eta\right] + \alpha S - \eta T_\lambda$ and the sum of marginal distribution and conditional distribution is "almost" 0, so we can say that it is a constant.

Q4 Collasped Metropolis-Within-Gibbs MCMC Sampler for Posterior Distribution

(1) Using MWG to generate (α,λ) targeting the marginal distribution $p(\alpha,\lambda \mid Y)$

(2) Conditional draw η from Gamma$(\hat{\kappa},\hat{\gamma})$

Q5 Numerical Verification of MCMC Algorithm

1 Simulate Y

```r
set.seed(2018)
n <- 500
alpha <- 2.3
beta <- 1.7
lambda <- 0.26
y <- rggamma(n, alpha, beta, lambda)
```

2 Compute Marginal Distributions from ggamma.logmarg

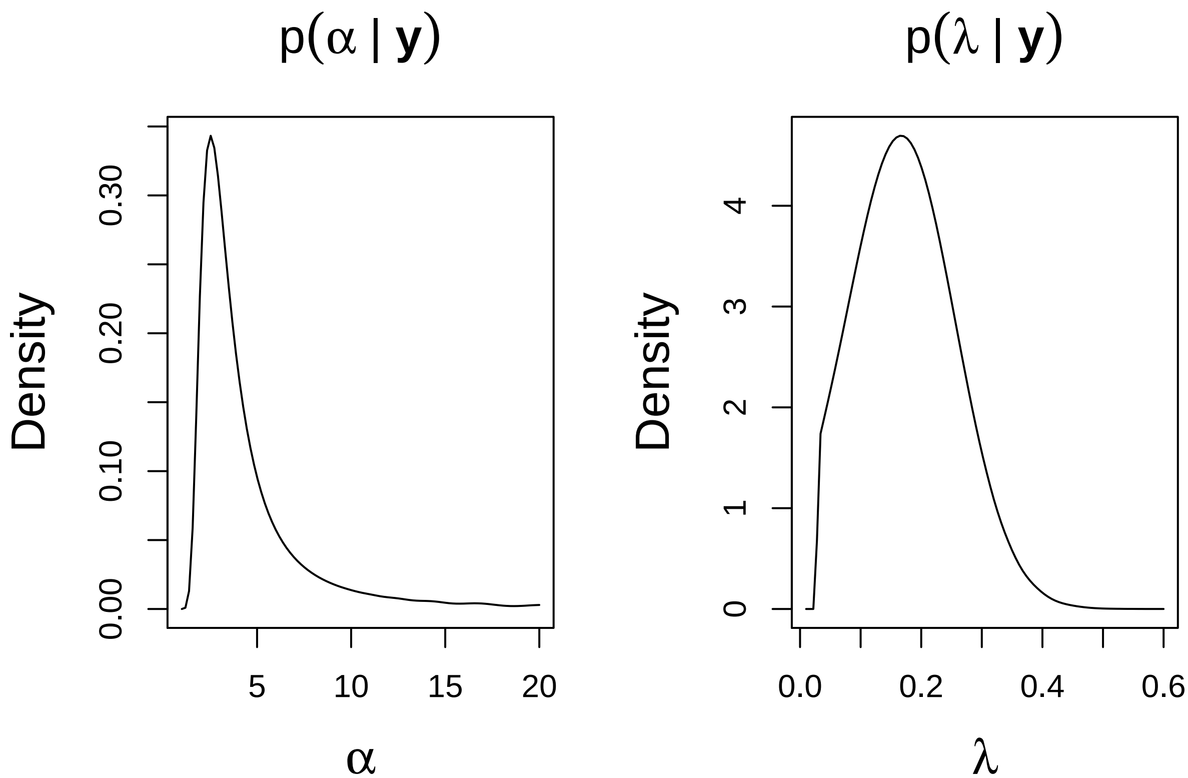
```r
npts <- 100
aseq <- seq(1, 20, len = npts)
lseq <- seq(0.010, 0.6, len = npts)
alseq <- as.matrix(expand.grid(aseq, lseq))
logy <- log(y)

# compute joint probability
lpmat <- apply(alseq, 1, function(theta) {
  khat <- n*theta[1]/theta[2]+1
  ghat <- sum(exp(theta[2]*logy))
  ggamma.logmarg(theta[1], theta[2], logy, khat, ghat) + logprior(theta[1], theta[2])
})
lpmat <- matrix(lpmat, npts, npts)

# compute marginal probability
aldens <- exp(lpmat-max(lpmat))
adens <- rowSums(aldens)
da <- aseq[2]-aseq[1]
adens <- adens/sum(adens)/da # normalize

ldens <- colSums(aldens)
dl <- lseq[2]-lseq[1]
ldens <- ldens/sum(ldens)/dl

# plot marginal distributions
cex <- 1.5
clrs <- c("black", "blue", "red")
par(mfrow = c(1,2))
plot(x = 0, type = "n", xlim = range(aseq), ylim=range(adens),
     cex.lab = cex, cex.main = cex,
     xlab = expression(alpha), ylab = "Density",
     main = expression(p(alpha*" | "*bold(y))))
lines(aseq, adens, col = clrs[1])
plot(x = 0, type = "n", xlim = range(lseq), ylim = range(ldens),
     cex.lab = cex, cex.main = cex,
     xlab = expression(lambda), ylab = "Density",
     main = expression(p(lambda*" | "*bold(y))))
lines(lseq, ldens, col = clrs[1])
```



3 Manually Tune the MWG Jump Sizes to Get 45% Acceptance Rate

```
set.seed(2018)
alpha0 <- rlnorm(1,0,2)
lambda0 <- rlnorm(1,0,2)
M <- 500
ggamma.post(nsamples = M, y = y, alpha0 = alpha0, lambda0 = lambda0,
            mwg.sd = c(0.505,0.025), acc.out = TRUE)$accept
```

```
## MH Acceptance Rate:

## gamma: 46%

## lambda: 47%

## eta: 68%

##     alpha    lambda       eta
## 0.4563636 0.4727273 0.6818182
```

After a few try on $M = 500$, choose `mwg.sd` $= (0.505, 0.025)$ as the jump size.

4 MCMC Sampling for M = 200000 Iterations

```
set.seed(2018)
alpha0 <- rlnorm(1,0,2)
lambda0 <- rlnorm(1,0,2)
M <- 200000
```

```r
result <- ggamma.post(nsamples = M, y = y, alpha0 = alpha0, lambda0 = lambda0,
                      mwg.sd = c(0.505,0.025), acc.out = TRUE)
```

```
## MH Acceptance Rate:

## gamma: 47%

## lambda: 45%

## eta: 68%
```

```r
result$accept
```

```
##     alpha    lambda       eta
## 0.4706119 0.4497512 0.6825622
```

5 Plot the Histograms of MCMC vs. Marginal Distributions

```r
par(mfrow = c(1,2))
hist(result$Theta[,1],breaks=100, freq=FALSE,
     xlab = expression(alpha),
     main = expression(p(alpha*" | "*bold(y))))
lines(aseq, adens, col = clrs[2])
hist(result$Theta[,2],breaks=100, freq=FALSE,
     xlab = expression(lambda),
     main = expression(p(lambda*" | "*bold(y))))
lines(lseq, ldens, col = clrs[2])
```