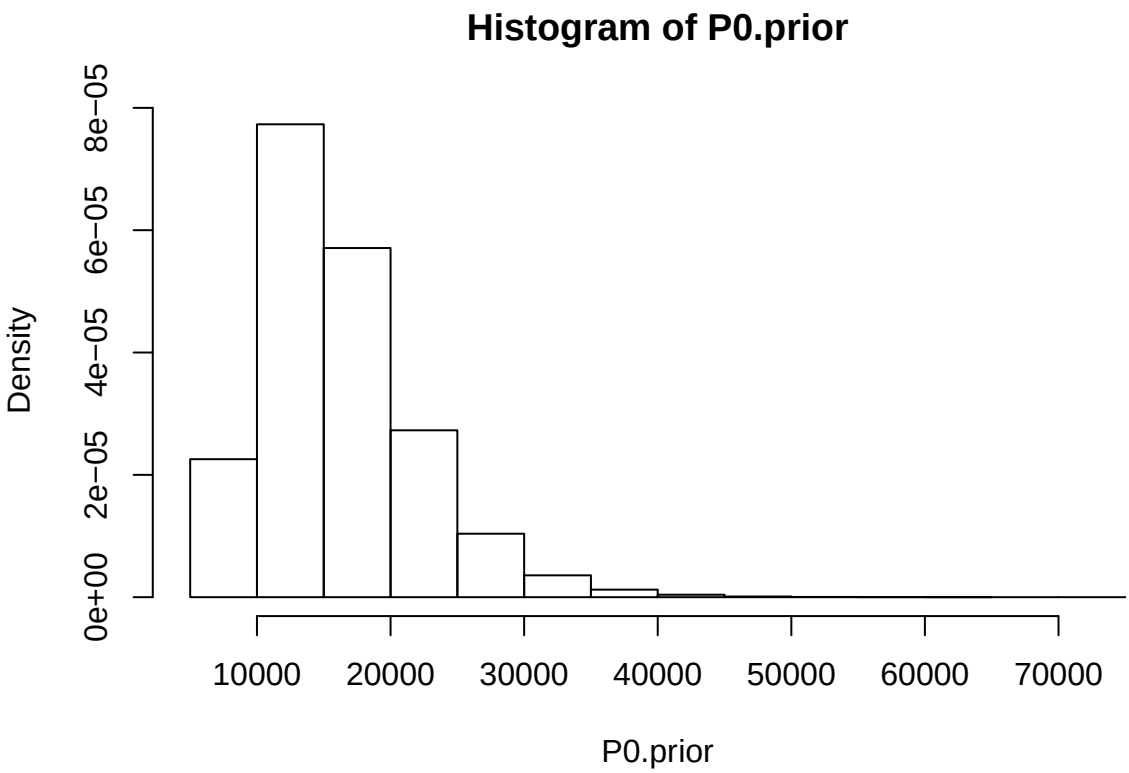
bowead_example.r

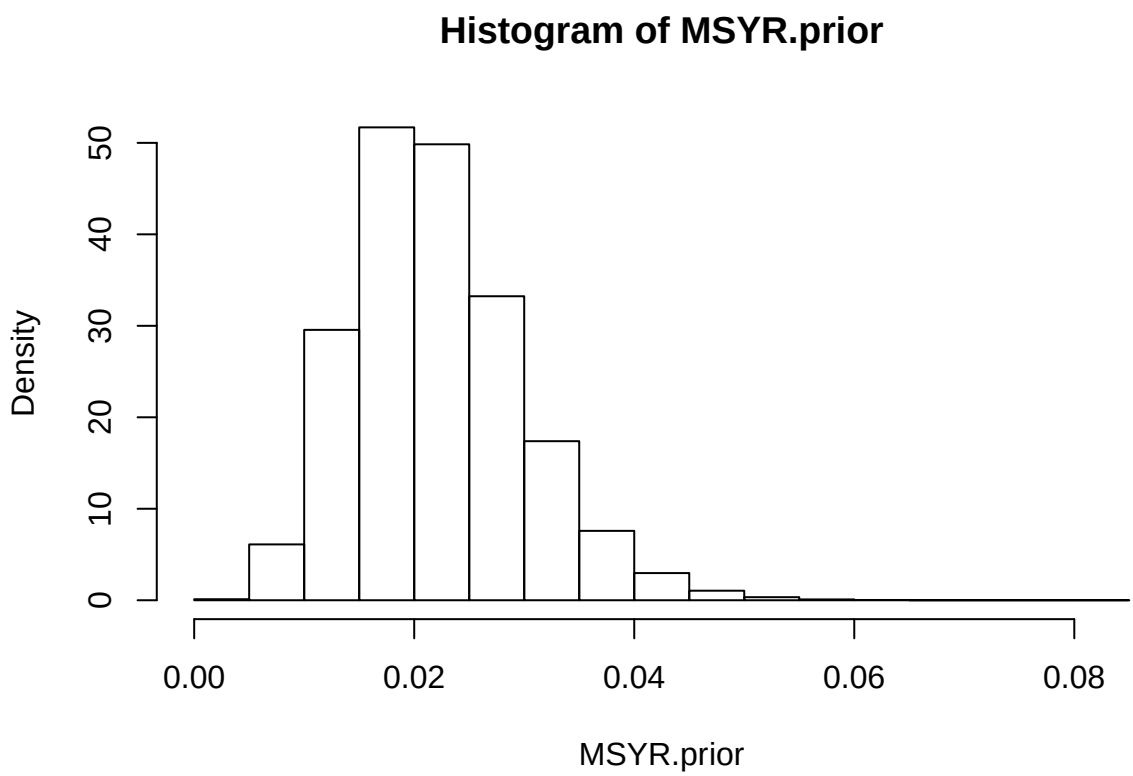
max

Fri Jan 27 14:43:28 2017

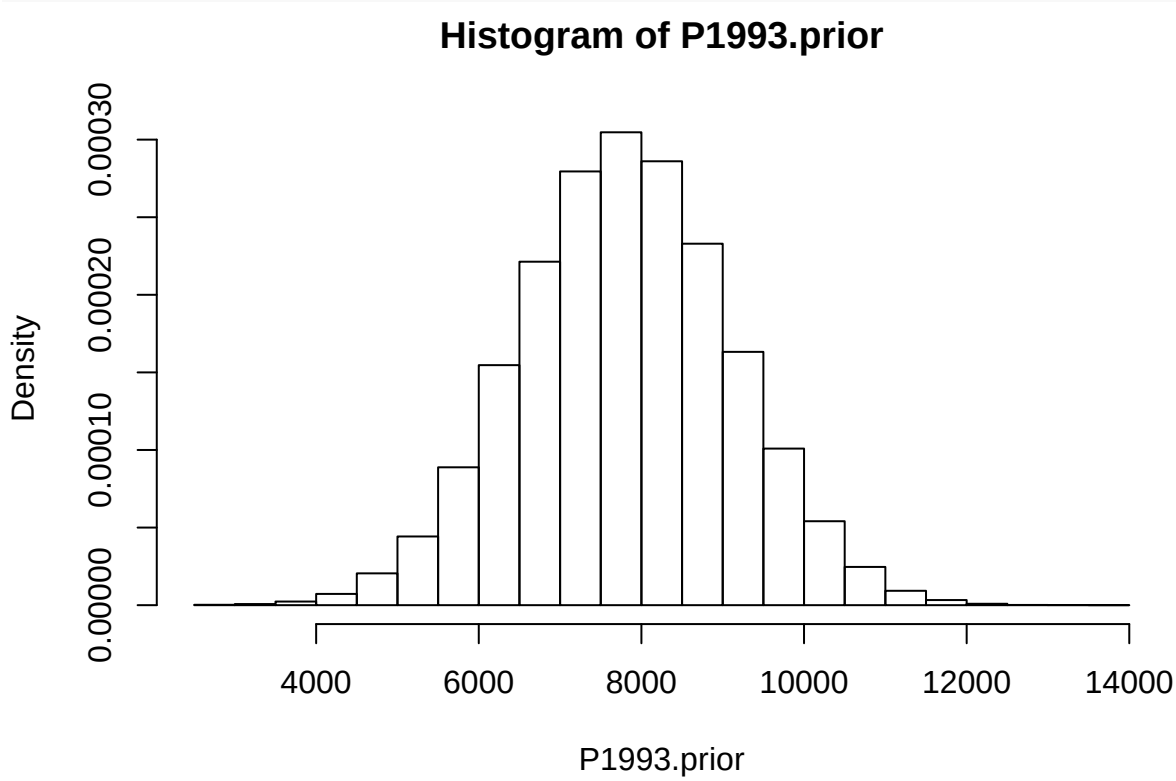
```r
# P_{t+1} = P_t - C_t + 1.5 * (MSYR) * P_t(1-(P_t/P_0)^2)
# Inputs: P_0 and MSYR
# Output: P_{1993}
source("pooling_aux.r")
## Prior for P_0 is a Gamma(2.8085, 0.0002886) + 6400, i.e, with offset 6400.
M <- 1E5
P0.prior <- rgamma(M, shape = 2.8085, rate = 0.0002886) + 6400
hist(P0.prior, probability = TRUE)
```



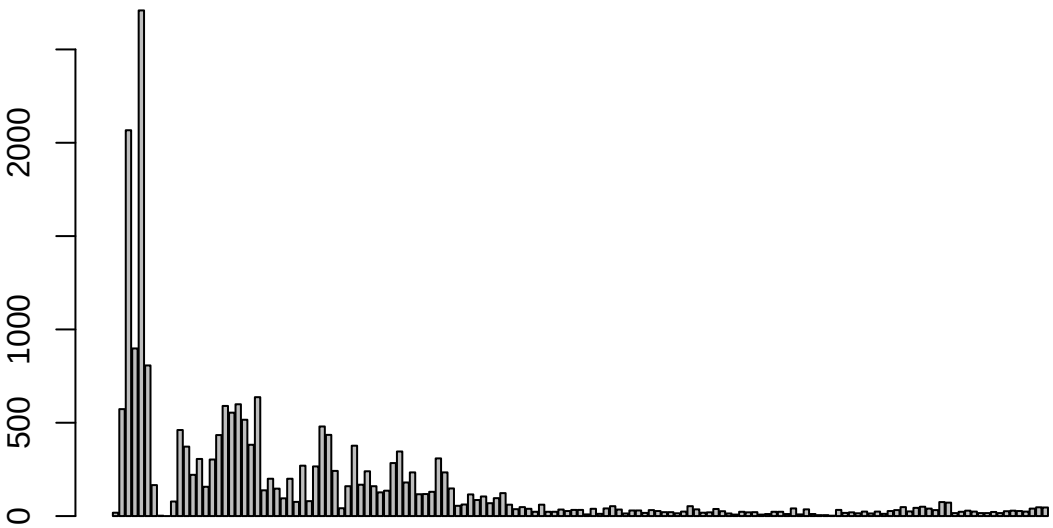

```r
## Prior for MSYR is Gamma(8.2, 372.7)
MSYR.prior <- rgamma(M, shape = 8.2, rate = 372.7)
hist(MSYR.prior, probability = TRUE)
```



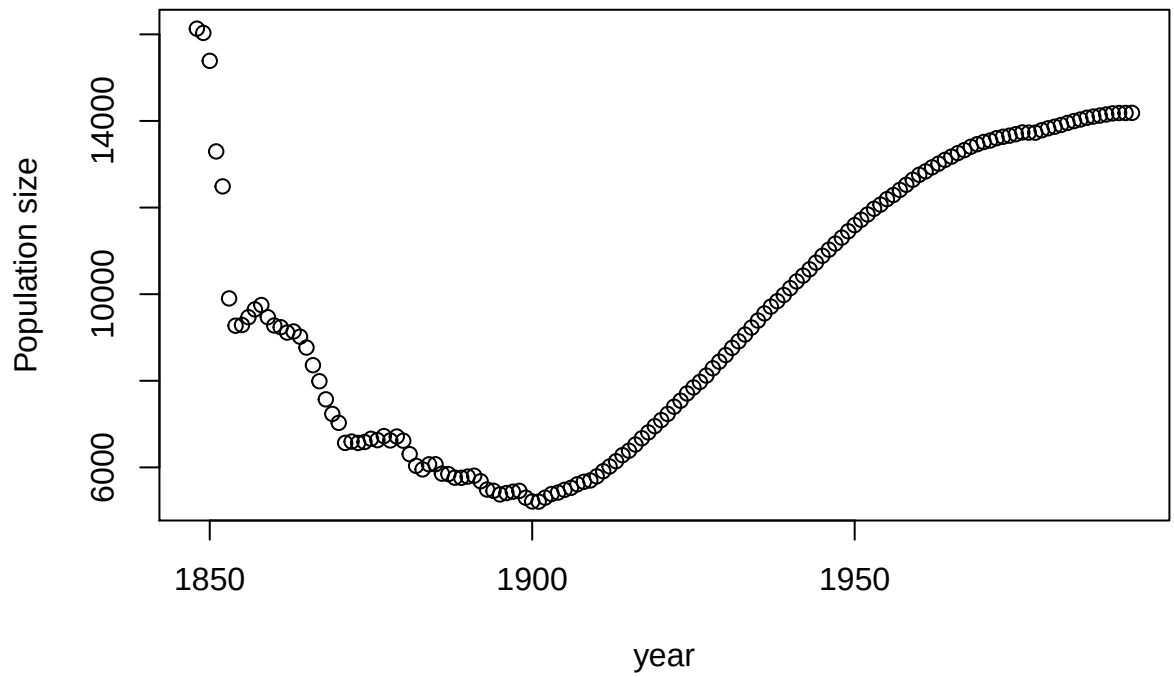

```
## Prior on P_{1993} is N(7800, 1300^2)
P1993.prior <- rnorm(M, mean = 7800, sd = 1300) ## q_2(\phi)
hist(P1993.prior, probability = TRUE)
```



```r
P1993.prior.dens <- density(P1993.prior)

P1993.likelihood2 <- function(x) dnorm(x, m = 8293,  sd = 626) ## L_2(\phi)

CtData <- read.csv("../data/bowhead_kills_1848-2002.csv")
CtData <- subset(CtData, year <= 1992)
barplot(CtData$kills)
```




```r
getP <- function(msyr, p0, C, Pinit){
  TT <- length(C)
  P <- rep(NA, TT + 1)
  P[1] <- Pinit
  for(t in 1: TT)  P[t+1] <- P[t] - C[t] + 1.5 * msyr * P[t]*(1-(P[t]/p0)^2)
  return(P)
}
plot(
  seq(min(CtData$year), max(CtData$year) + 1, by = 1),
  getP(msyr = median(MSYR.prior),
       p0 = median(P0.prior),
       C = CtData$kills,
       Pinit = mean(P0.prior)),
  xlab = "year", ylab = "Population size"
)
```



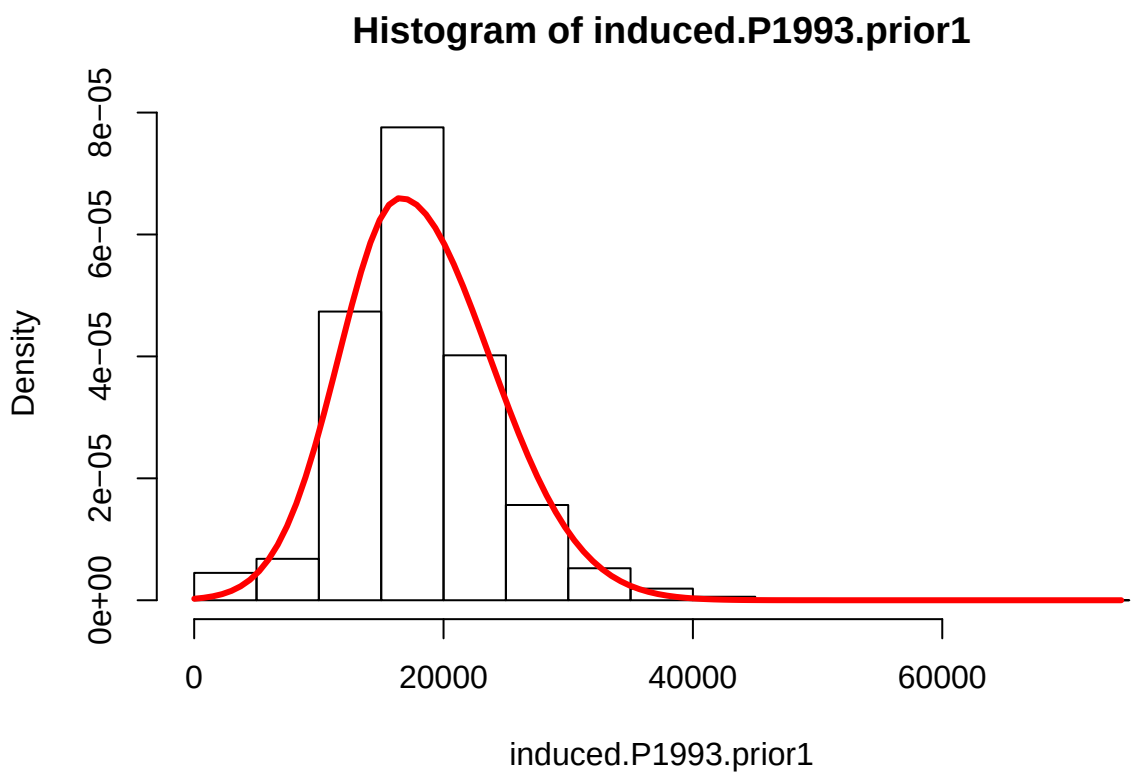

```
induced.P1993.prior1 <- unlist(
  parallel::mclapply(1:M, function(i){
    simuP <- getP(msyr = MSYR.prior[i],
                  p0 = P0.prior[i],
                  C = CtData$kills,
                  Pinit = P0.prior[i])
    return(simuP[length(simuP)])
  }, mc.cores = 4)
)
induced.P1993.prior1 <- induced.P1993.prior1[induced.P1993.prior1>0]
induced.P1993.prior1.dens <- density(induced.P1993.prior1)

summary(induced.P1993.prior1)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     8.92 14490.00 17420.00 18150.00 21300.00 74350.00
```

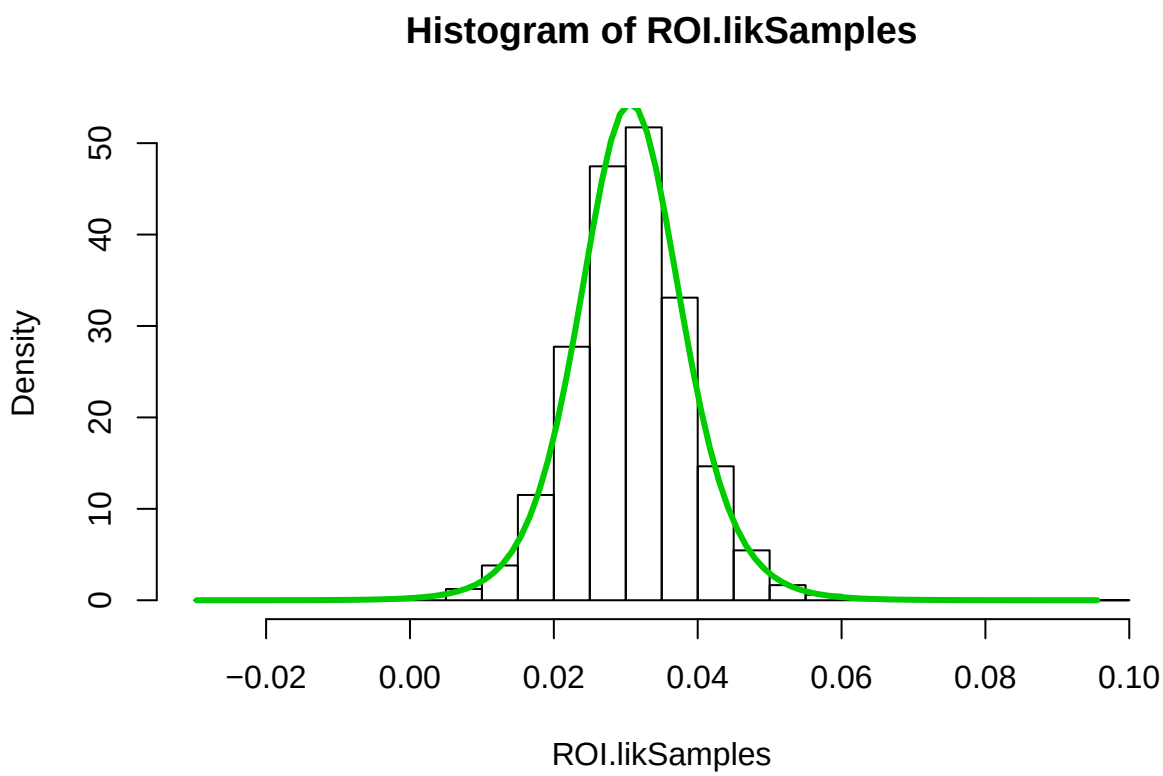
```
c(mean(induced.P1993.prior1), quantile(induced.P1993.prior1, probs = c(.025, .5, .975)))
```

```
##                2.5%       50%     97.5%
## 18149.030  5529.627 17418.470 32141.866
```

```
hist(induced.P1993.prior1, probability = TRUE)
distFit <- fGarch::snormFit(induced.P1993.prior1)
curve(fGarch::dsnorm(x, mean = distFit$par["mean"],
                     sd = distFit$par["sd"], xi =   distFit$par["xi"]),
      min(induced.P1993.prior1), max(induced.P1993.prior1), col = 2, lwd = 3, add = TRUE)
```




```
###########
# Likelihood for ROI
# P1993  = P1978* (1 + ROI)^15
# Likelihood(ROI) = y = g(t8) = exp(a + bt8) -1 where
# t8 is a student t-distribution with df = 8
# g_inverse(y) = t8 = (log(y + 1) - a)/b
# dg_inverse(y)/dy = 1/b(y + 1), hence |J| = abs(1/b(y + 1))
# here a = .0302 and b = .0069
ROI.likelihood <- function(y, a = .0302, b = .0069, log = FALSE){
  if(is.nan(y) && log == FALSE) return(0)
  if(is.nan(y) && log == TRUE) return(-Inf)
  ## L_2(\phi)
  if(log){
    dt((log(y + 1) - a)/b, df = 8, log = TRUE) + log(abs(1/(b * (y + 1))))
  }else{
    dt((log(y + 1) - a)/b, df = 8) *  abs(1/(b * (y + 1)))
  }
}
ROI.likSamples <- exp(.0302  + .0069 * rt(M, df = 8)) - 1

hist(ROI.likSamples, probability = TRUE)
curve(ROI.likelihood, min(ROI.likSamples), max(ROI.likSamples),
      lwd = 3, col = 3, add = TRUE)
```

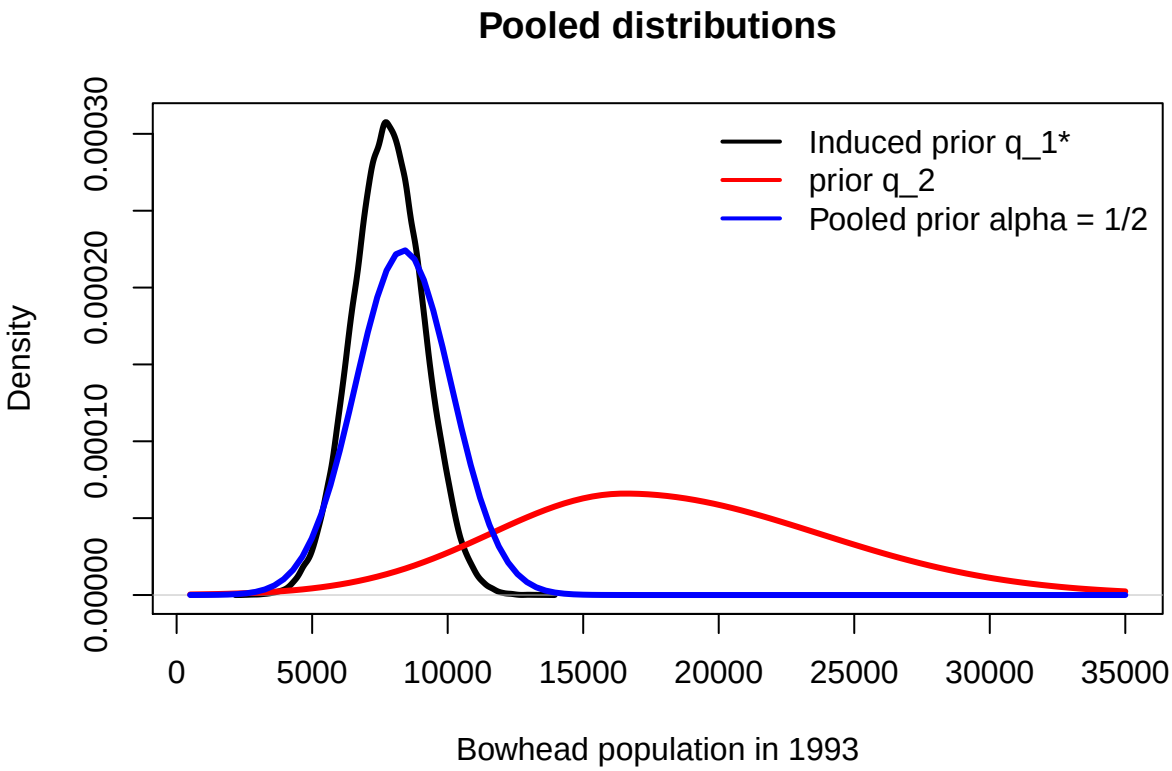



```
getROI <- function(pars){
  Ps <-  getP(msyr = pars[2], p0 = pars[1],
              C = CtData$kills, Pinit = pars[1])
  P1978 <- Ps[length(Ps) - 15]
  P1993 <- Ps[length(Ps)]
  ROI <- (P1993/P1978)^{1/15} - 1
  return(ROI)
}
#########
# The pooled prior LP({q_2(\phi), q_1^\ast(\phi)}, alpha),
P1993.pooledPrior <- function(x, weights){
  if(is.nan(x)) return(0)
  dpoolnorm(x = x,
            D = list(
              f0 = function(x) dnorm(x, mean = 7800, sd = 1300),
              f1 = function(x) fGarch::dsnorm(x, mean = distFit$par["mean"],
                                              sd = distFit$par["sd"],
                                              xi =   distFit$par["xi"])),
            alphas = weights)
}
##########
Alpha <- .5
alphas <- c(Alpha, 1-Alpha)
###
plot(P1993.prior.dens, lwd = 3, xlim = c(500, 35000),
     main = "Pooled distributions", xlab = "Bowhead population in 1993")
curve(fGarch::dsnorm(x, mean = distFit$par["mean"],
                     sd = distFit$par["sd"], xi = distFit$par["xi"]),
      500, 35000, col = 2, lwd = 3, add = TRUE)
```

```r
curve(P1993.pooledPrior(x, weights = c(Alpha, 1-Alpha) ),
      500, 35000,
      col = "blue", lwd = 3, add = TRUE)
```

```
## Warning in if (is.nan(x)) return(0): the condition has length > 1 and only
## the first element will be used
```

```r
legend(x = "topright",
       legend = c("Induced prior q_1*", "prior q_2", "Pooled prior alpha = 1/2"),
       col = c("black", "red", "blue"), lwd = 2, bty = "n")
```




```r
#########
# Now let's get the posterior using LP-SpIR [for fixed alpha]
getP1993 <- function(theta){
  P <- getP(msyr = theta[2], p0 = theta[1],
            C = CtData$kills, Pinit = theta[1])
  return(P[length(P)])
}
SpIR.posterior <- LP.SpIR(
  k = M,
  l = round(.2 * M),
  Model = getP1993,
  rq1 = function(n) cbind(rgamma(n, shape = 2.8085, rate = 0.0002886) + 6400,
                          rgamma(n, shape = 8.2, rate = 372.7)),
  dq2 = function(x) dnorm(x, m = 7800, sd = 1300),
  dL2 = P1993.likelihood2,
  dL1 = function(x) ROI.likelihood(getROI(x)),
  alpha = Alpha
)
```

```r
# Posterior for the input
Theta.posterior <- data.frame(SpIR.posterior$theta.resamp)
names(Theta.posterior) <- c("P0", "MSYR")
Theta.posterior <- data.frame(
  Theta.posterior,
  ROI = apply(Theta.posterior, 1, getROI)
)
#
Theta.prior <- data.frame(
  P0 = P0.prior,
  MSYR = MSYR.prior,
  ROI = ROI.likSamples
)
apply(Theta.prior, 2,
      function(x) c(mean = mean(x), quantile(x, probs = c(.025, .5, .975))))
```

```
##              P0        MSYR        ROI
## mean  16130.723 0.022028976 0.03068041
## 2.5%   8275.094 0.009584452 0.01432770
## 50%   15007.512 0.021144517 0.03068309
## 97.5% 30293.354 0.039409213 0.04724805
```

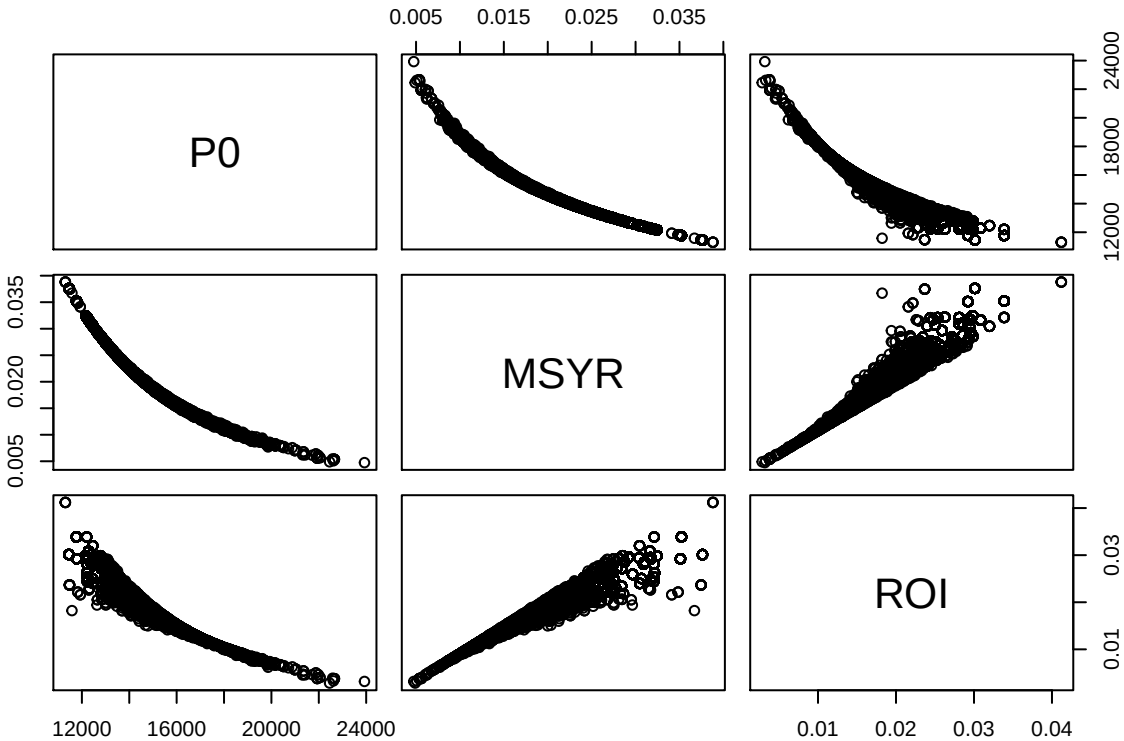
```r
apply(Theta.posterior, 2,
      function(x) c(mean = mean(x), quantile(x, probs = c(.025, .5, .975))))
```

```
##             P0       MSYR        ROI
## mean  14519.38 0.02136255 0.02013886
## 2.5%  12162.32 0.01137370 0.01052870
## 50%   14409.21 0.02077711 0.01984976
## 97.5% 17905.14 0.03244841 0.02984583
```

```r
cor(Theta.posterior)
```

```
##              P0       MSYR        ROI
## P0    1.0000000 -0.9709926 -0.9634029
## MSYR -0.9709926  1.0000000  0.9713268
## ROI  -0.9634029  0.9713268  1.0000000
```

```r
plot(Theta.posterior)
```




```r
# Now the output
P1993.posterior <- SpIR.posterior$phi.resamp
c(mean = mean(P1993.posterior), quantile(P1993.posterior, probs = c(.025, .5, .975)))
```

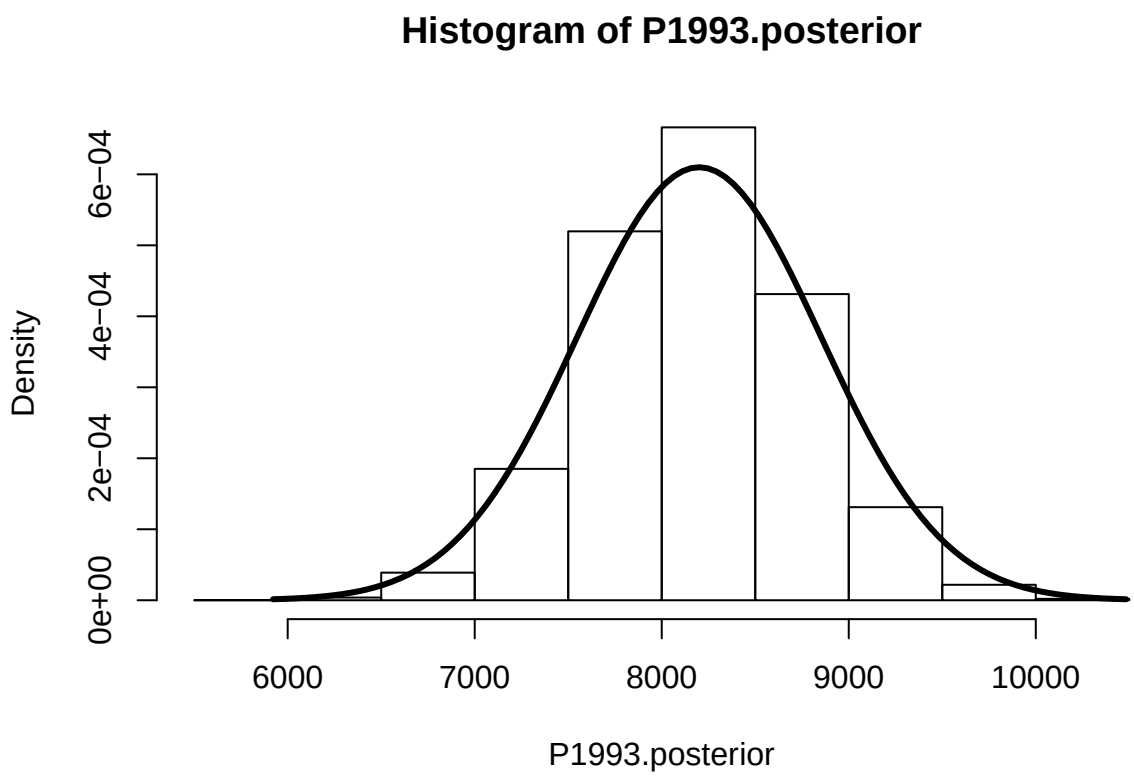
```
##     mean     2.5%      50%    97.5%
## 8183.078 7039.333 8209.631 9323.494
```

```r
c(mean = mean(P1993.prior), quantile(P1993.prior, probs = c(.025, .5, .975)))
```

```
##      mean      2.5%       50%     97.5%
##  7800.517  5259.859  7794.498 10348.458
```

```r
hist(P1993.posterior, probability = TRUE)
curve(dnorm(x, mean = 8200, sd = 654),
      min(P1993.posterior), max(P1993.posterior),
      add = TRUE, lwd = 3)

###########################
###########################
# Now let's approximate the posterior of theta using a different method
## Priors
prior <- function(param, X){
  npar <- length(param)
  K <- length(X)
  aalphas <- arm::invlogit(param[(npar-K +1):npar])
  dgamma(param[1] - 6400, shape = 2.8085, rate = 0.0002886, log = TRUE) +
    dgamma(param[2], shape = 8.2, rate = 372.7, log = TRUE)  +
    log(P1993.pooledPrior(getP1993(param[1:2]), weights = aalphas)) +
    log(gtools::ddirichlet(aalphas, alpha = X))
}
## Likelihoods
L2 <- function(x){
```

```r
  if(is.nan(x)) return(-Inf)
  dnorm(x, m = 8293,  sd = 626, log = TRUE)
}
Lik <- function(param){
  # L1(\theta) * L2(M(\theta))
  L2(getP1993(theta = param[1:2]) ) +
    ROI.likelihood(getROI(pars = param[1:2]), log = TRUE)
}
## Posterior
posterior <- function(param, X, verbose = FALSE){
  Lk <- Lik(param)
  pr <-  prior(param, X = X)
  if(verbose) cat("Likelihood:", Lk, " prior:", pr,
                  " P0:", param[1], " MSYR:", param[2],
                  " alphas:", arm::invlogit(param[3:4]),
                  "\n")
  return (Lk + pr)
}
## Compile stuff to try and improve performance
prior <- compiler::cmpfun(prior)
L2 <- compiler::cmpfun(L2)
Lik <- compiler::cmpfun(Lik)
posterior <- compiler::cmpfun(posterior)
#
proposalfunction <- compiler::cmpfun(proposalfunction)
run_metropolis_MCMC <- compiler::cmpfun(run_metropolis_MCMC)
##
# Let's run the stuff
Niter <- 2 * M
system.time(
  Chain <- LPMCMC(startvalue = c(mean(P0.prior), mean(MSYR.prior), arm::logit(alphas)),
                  mm = c(mean(P0.prior), mean(MSYR.prior)),
                  SS = 10 * var(Theta.posterior[, 1:2]),
                  XX = rep(1/2, length(alphas)), ## Jeffreys prior
                  hh = c(.2, .2),
                  Nit = Niter, burnin = round(.5 * Niter))
)
```



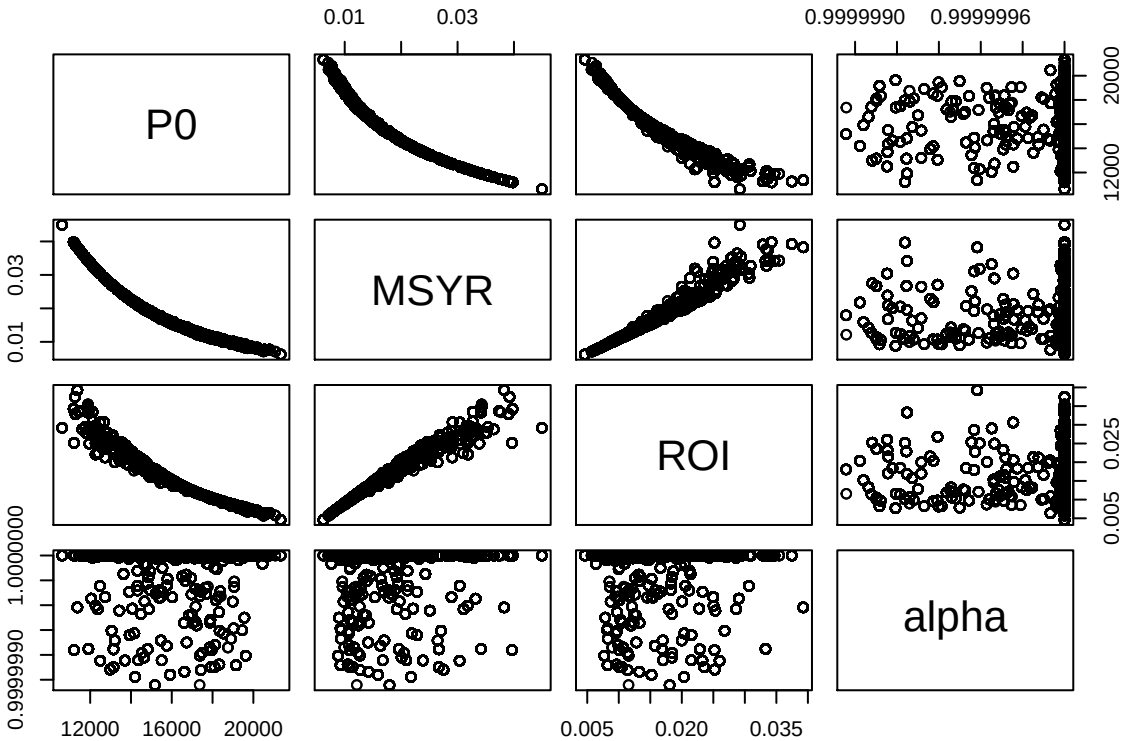

```
## [1] "iteration 20000 of 2e+05"
## [1] "iteration 40000 of 2e+05"
## [1] "iteration 60000 of 2e+05"
## [1] "iteration 80000 of 2e+05"
## [1] "iteration 100000 of 2e+05"
## [1] "iteration 120000 of 2e+05"
## [1] "iteration 140000 of 2e+05"
## [1] "iteration 160000 of 2e+05"
## [1] "iteration 180000 of 2e+05"
## [1] "iteration 200000 of 2e+05"
```

```
##      user    system   elapsed
## 21765.516    21.916 21800.339
```

```
Chain$acceptance
```

```
## [1] 0.005439946
```

```
MH_LP.Post <- data.frame(Chain$trace[, 1:2],
                         ROI = apply(Chain$trace[, 1:2], 1, getROI),
                         alpha = arm::invlogit(Chain$trace[, 3]))
names(MH_LP.Post) <- c("PO", "MSYR", "ROI", "alpha")
plot(MH_LP.Post)
```



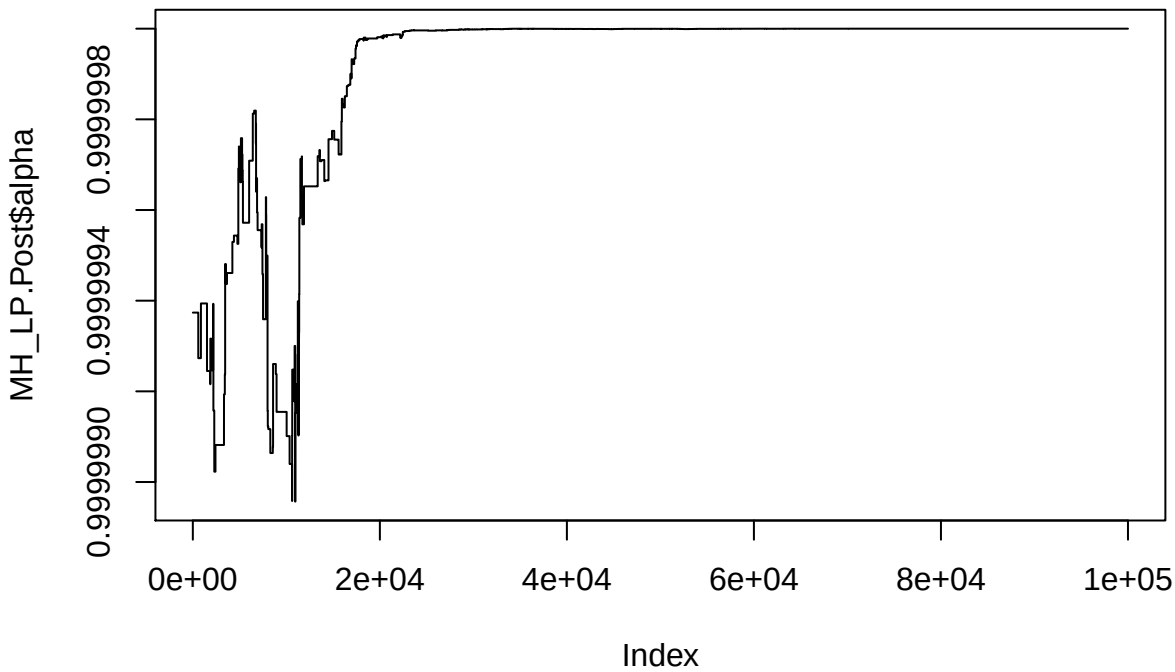

```r
apply(MH_LP.Post, 2, coda::effectiveSize)
```

```
##         P0       MSYR        ROI      alpha
## 209.734318 206.333871 194.836134   5.822145
```

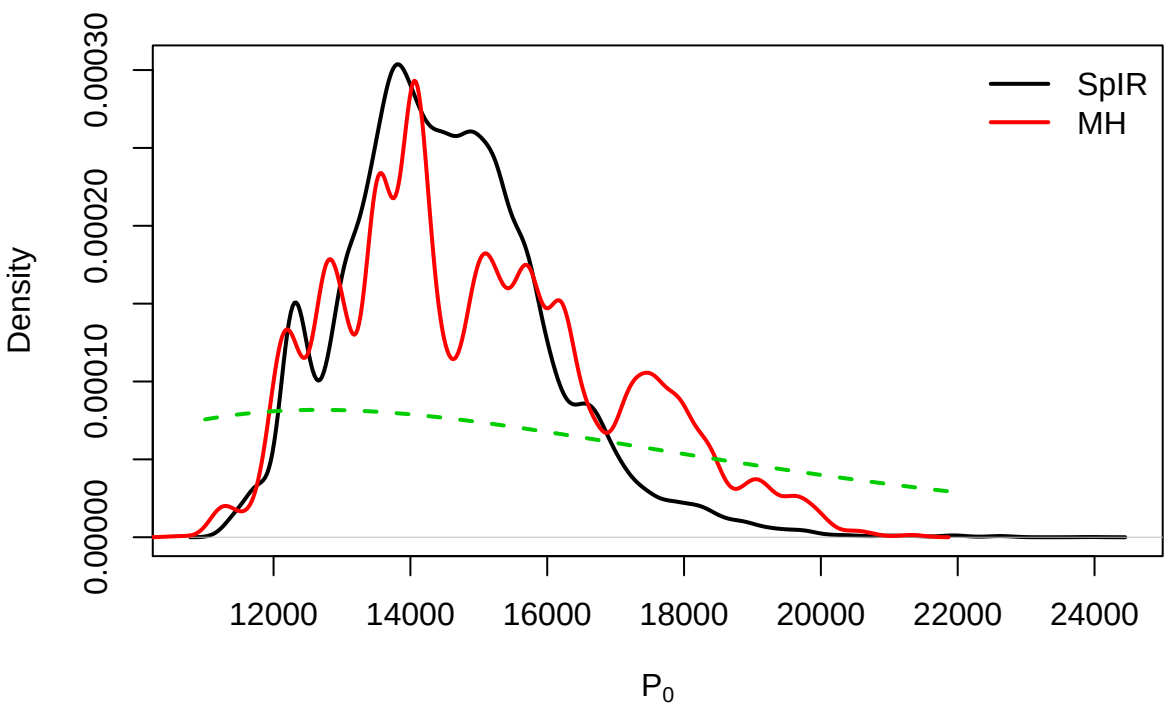
```r
apply(MH_LP.Post, 2,  function(x) c(mean = mean(x), quantile(x, probs = c(.025, .5, .975))))
```

```
##             P0        MSYR         ROI     alpha
## mean  14964.61 0.020315801 0.018948155 0.9999999
## 2.5%  12052.97 0.009266903 0.008223875 0.9999992
## 50%   14771.94 0.019613589 0.018321715 1.0000000
## 97.5% 19200.62 0.033200487 0.029633220 1.0000000
```

```r
plot(MH_LP.Post$alpha, type = "l")
```

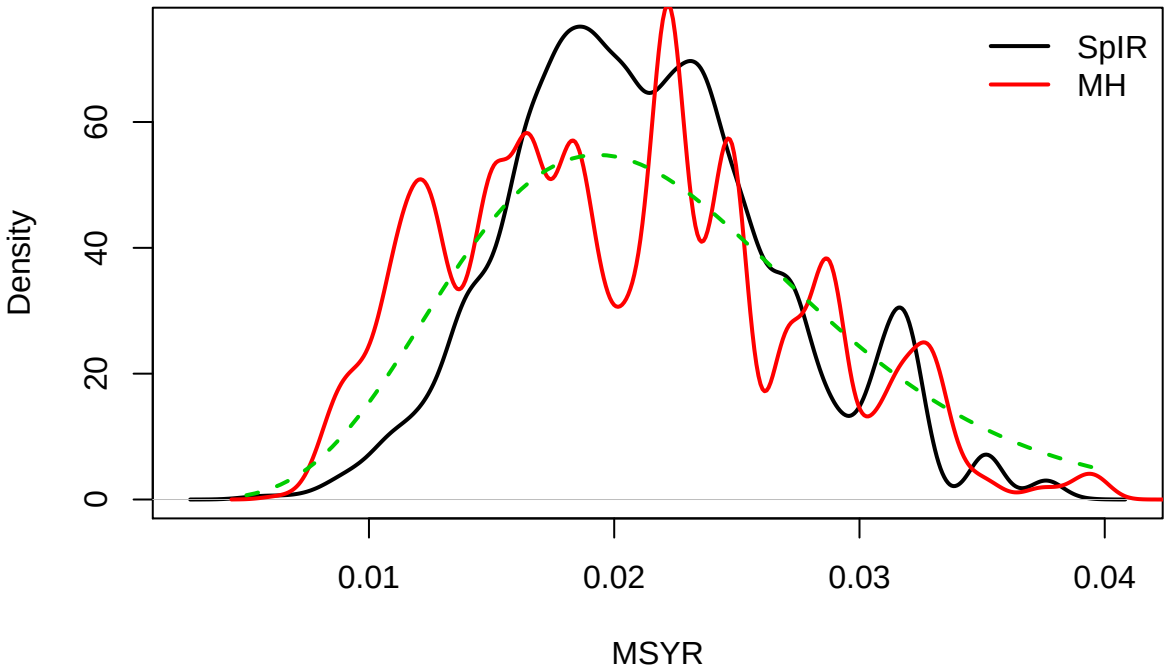


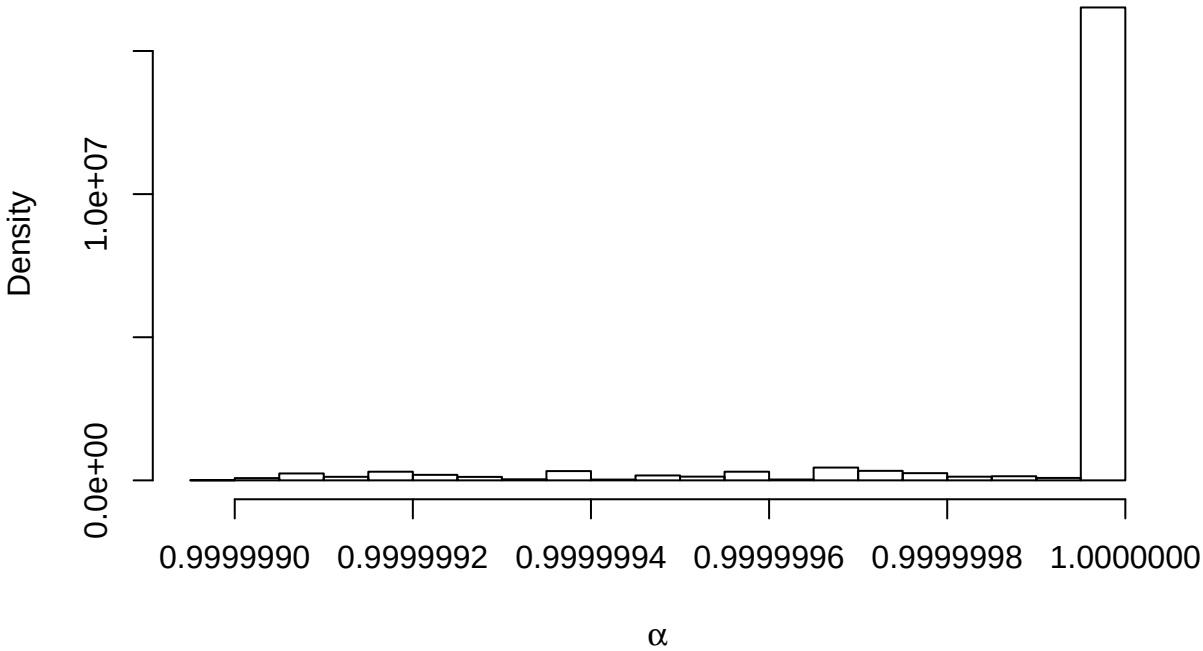

```
# Let's compare posteriors
plot(density(x = Theta.posterior[, 1]), main = "",
     lwd = 2, xlab = expression(P[0]))
lines(density(x = Chain$trace[, 1]), lwd = 2, col = 2)
curve(dgamma(x-6400, shape = 2.8085, rate = 0.0002886),
      11000, 22000,
      lwd = 2, lty = 2, col = 3, add = TRUE)
legend(x = "topright", legend = c("SpIR", "MH"), col = 1:2, lwd = 2, bty = "n")
```




```
#
plot(density(x = Theta.posterior[, 2]), main = "",
```

```
     lwd = 2, xlab = "MSYR")
lines(density(x = Chain$trace[, 2]), lwd = 2, col = 2)
curve(dgamma(x, shape = 8.2, rate = 372.7),
      0.005, 0.04,
      lwd = 2, lty = 2, col = 3, add = TRUE)
legend(x = "topright", legend = c("SpIR", "MH"), col = 1:2, lwd = 2, bty = "n")
```




```
#
hist(MH_LP.Post$alpha, probability = TRUE, main = "",  xlab = expression(alpha))
```