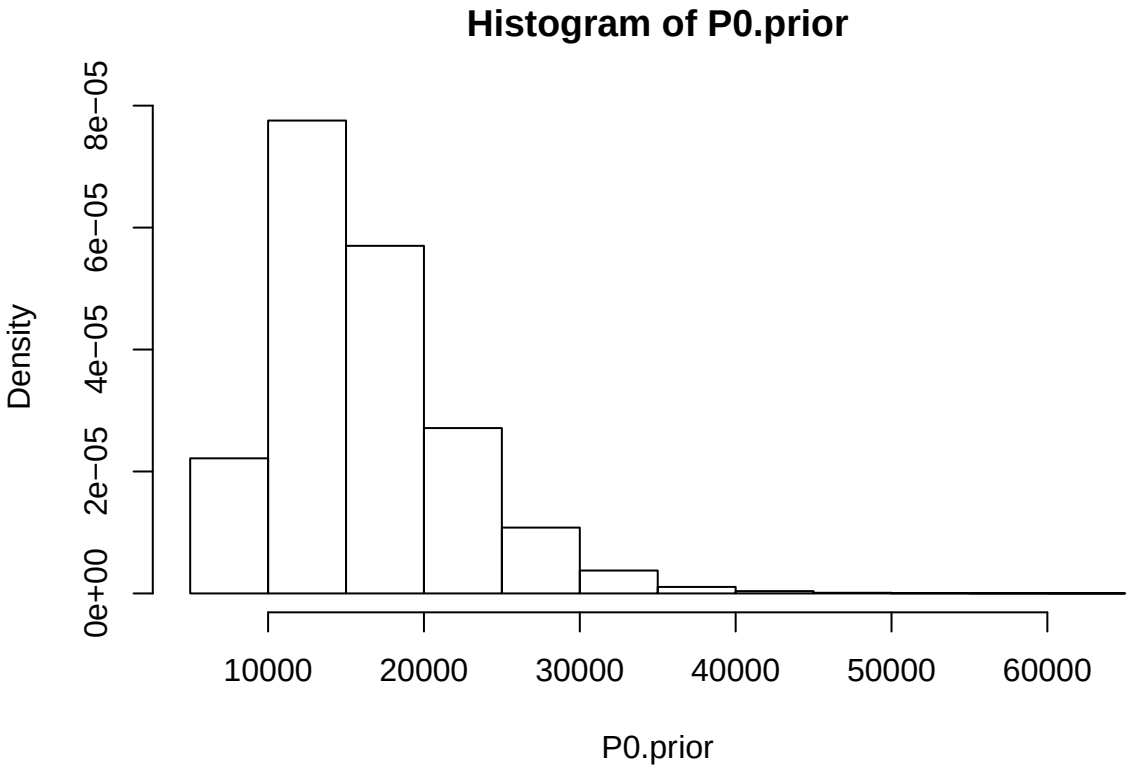
bowead_example_P0____MSYR.r

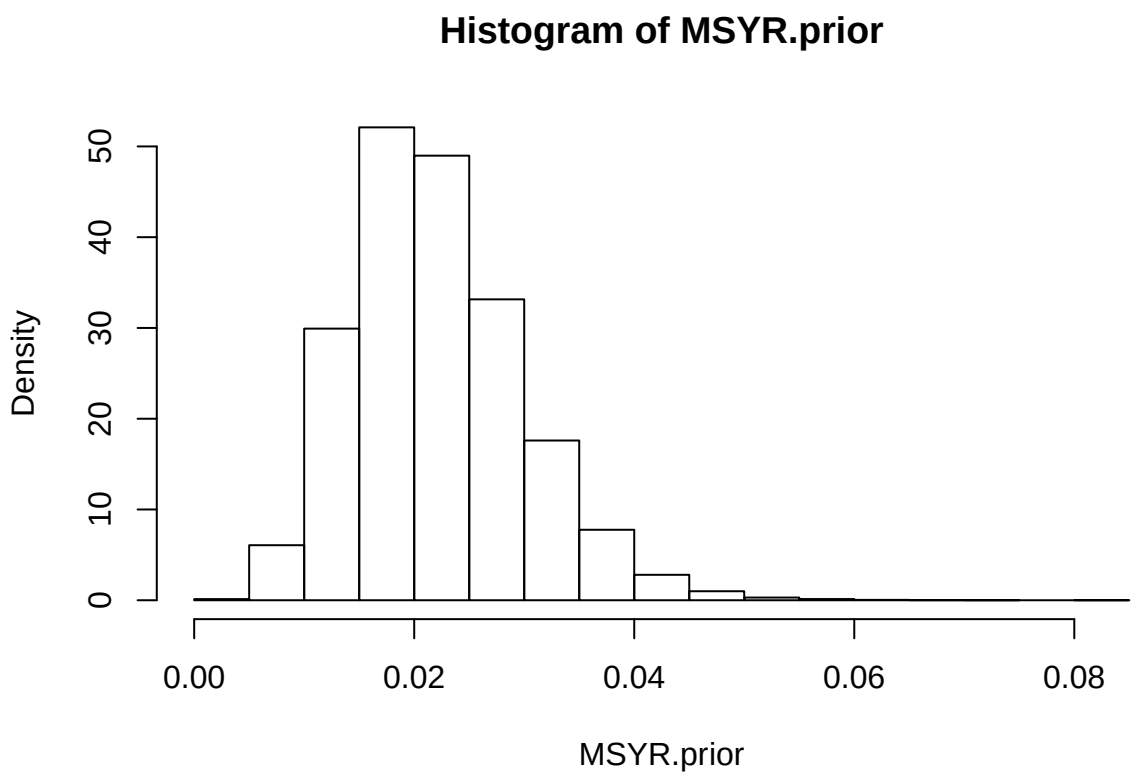
max

Sun Jan 22 00:51:30 2017

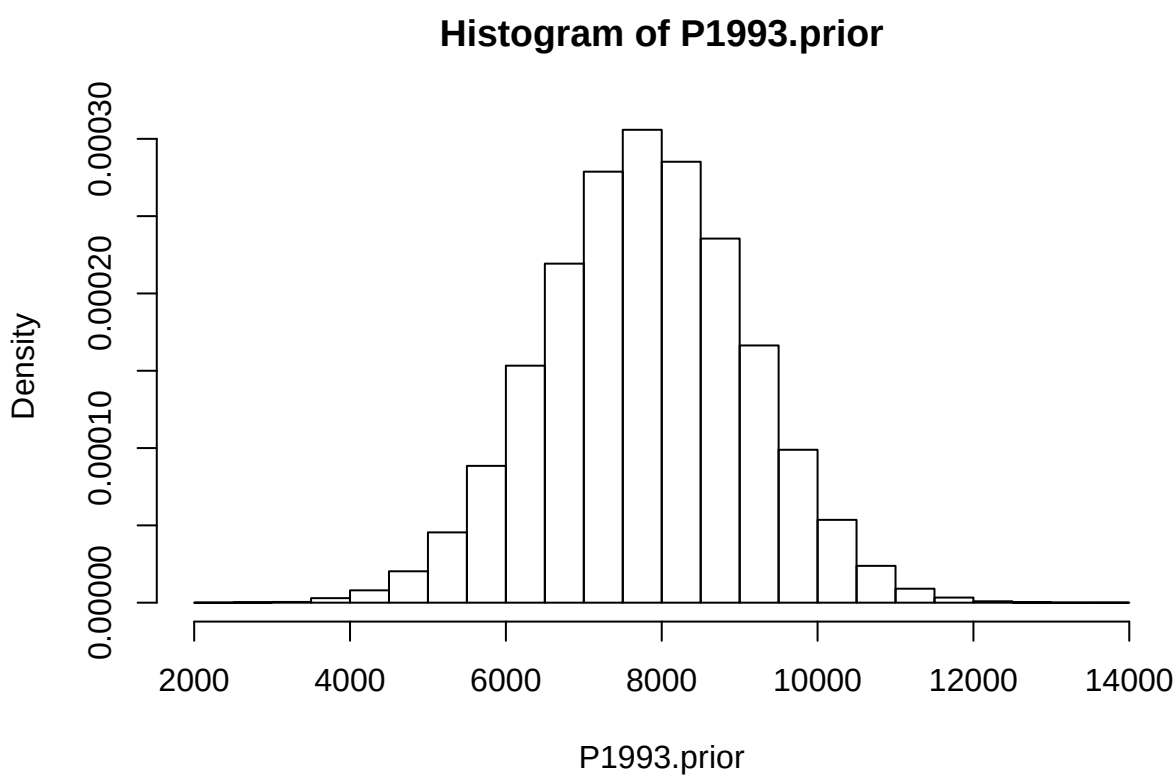
```
# P_{t+1} = P_t - C_t + 1.5 * (MSYR) * P_t(1-(P_t/P_0)^2)
# Inputs: P_0 and MSYR
# Output: P_{1993}
source("pooling_aux.r")
## Prior for P_0 is a Gamma(2.8085, 0.0002886) + 6400, i.e, with offset 6400.
M <- 1E5
P0.prior <- rgamma(M, shape = 2.8085, rate = 0.0002886) + 6400
hist(P0.prior, probability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



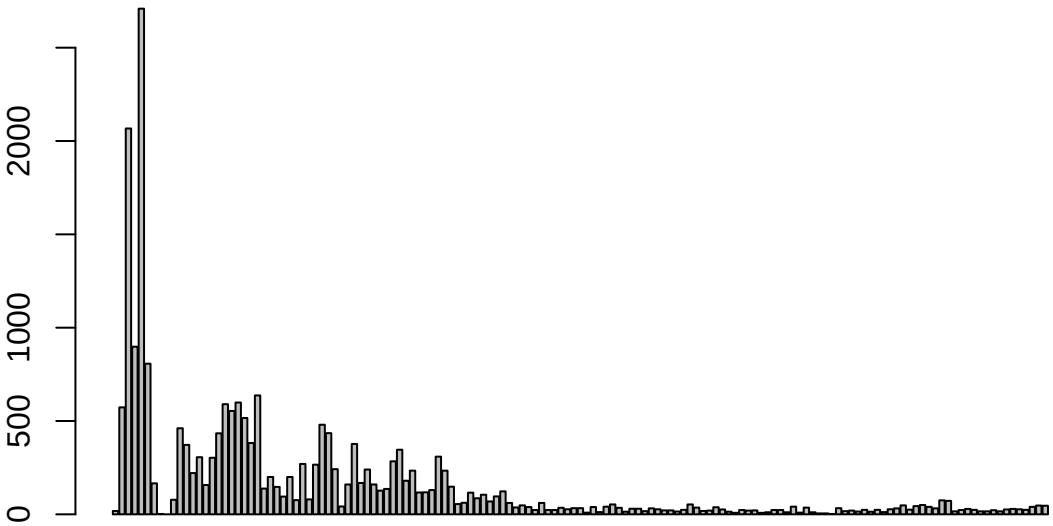

```
## Prior for MSYR is Gamma(8.2, 372.7)
MSYR.prior <- rgamma(M, shape = 8.2, rate = 372.7)
hist(MSYR.prior, probability = TRUE)
```



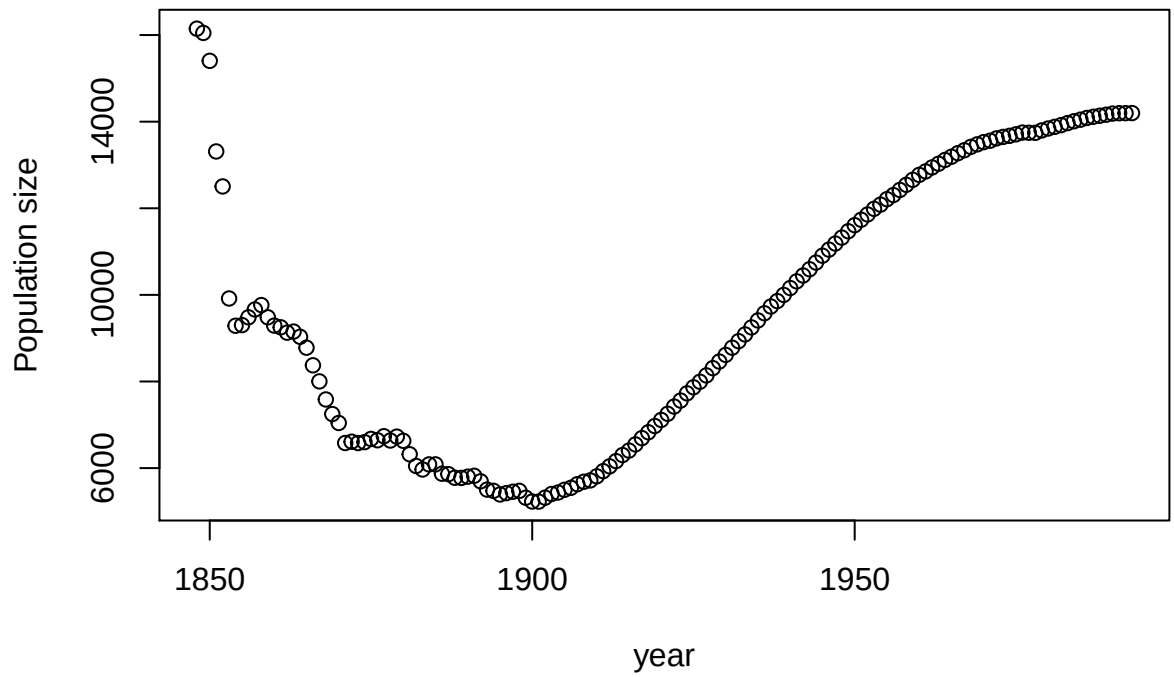

```
## Prior on P_{1993} is N(7800, 1300^2)
P1993.prior <- rnorm(M, mean = 7800, sd = 1300) ## q_2(\phi)
hist(P1993.prior, probability = TRUE)
```



```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"
)
```



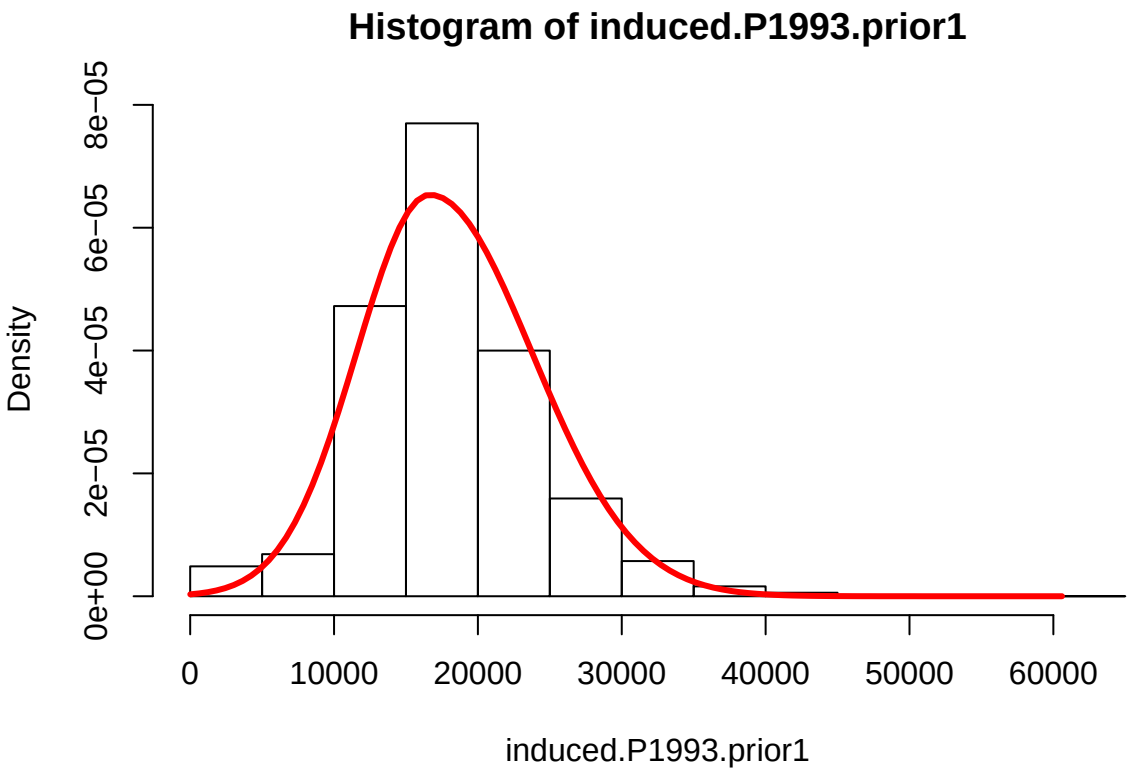

```
induced.P1993.prior1 <- unlist(
  parallel::mclapply(1:M, function(i){
    simuP <- getP(msyr = MSYR.prior[i],
                  p0 = P0.prior[i],
                  C = CtData$kills,
                  Pinit = P0.prior[i])
    return(simuP[length(simuP)])
  }, mc.cores = 4)
)
induced.P1993.prior1 <- 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.
##     7.76 14460.00 17430.00 18130.00 21330.00 60600.00
```

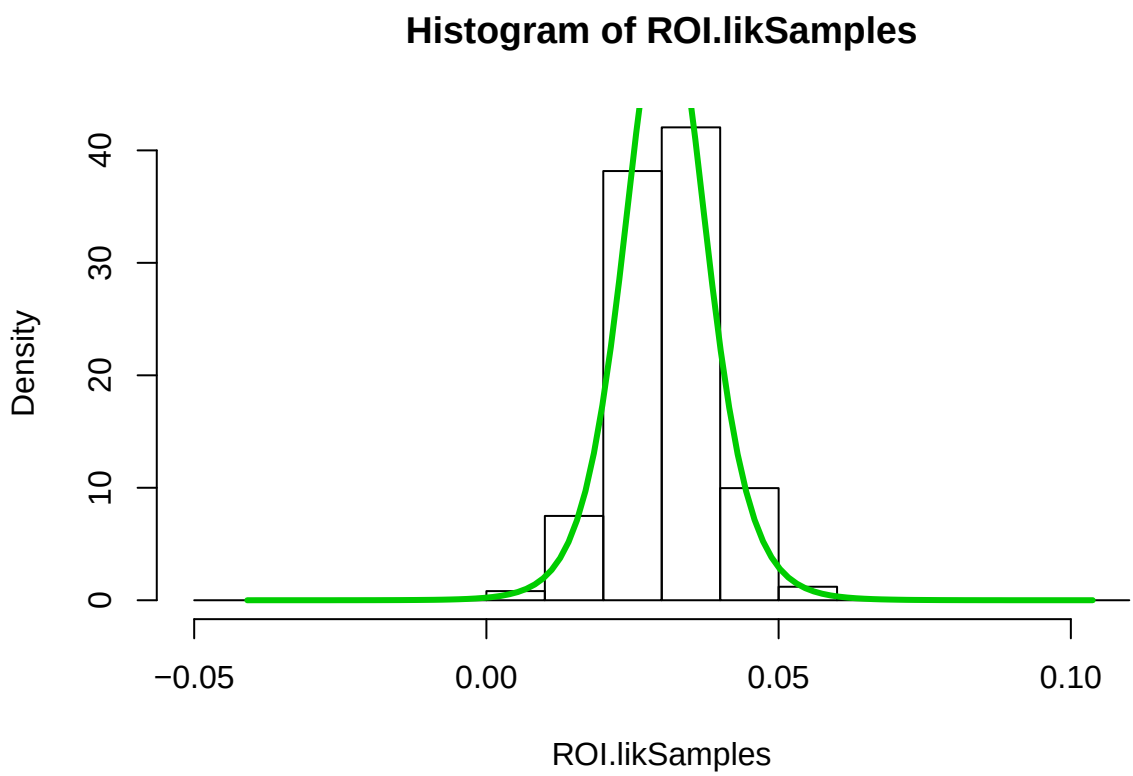
```
c(mean(induced.P1993.prior1), quantile(induced.P1993.prior1, probs = c(.025, .5, .975)))
```

```
##                2.5%       50%     97.5%
## 18132.793  5099.617 17429.328 32031.482
```

```
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)
```

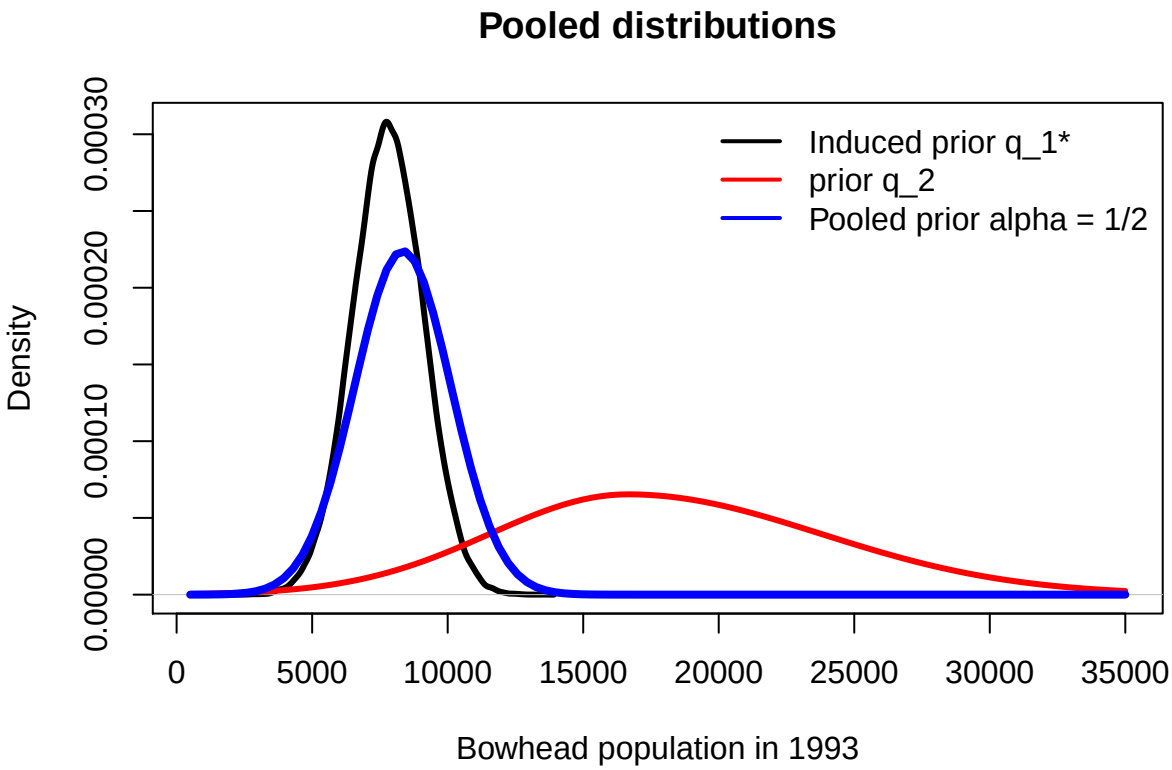



```
getROI <- function(pars){
  Ps <-  getP(msyr = pars[2], p0 = pars[1],
              C = CtData$kills, Pinit = pars[1])
  P1978 <- Ps[length(Ps) - 15]
  P1993 <- Ps[length(Ps)]
  ROI <- (P1993/P1978)^{1/15} - 1
  return(ROI)
}
#########
# The pooled prior LP({q_2(\phi), q_1^\ast(\phi)}, alpha),
P1993.pooledPrior <- function(x, weights){
  if(is.nan(x)) return(0)
  dpoolnorm(x = x,
            D = list(
              f0 = function(x) dnorm(x, mean = 7800, sd = 1300),
              f1 = function(x) fGarch::dsnorm(x, mean = distFit$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 = 4, 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  16146.893 0.022005038 0.03070164
## 2.5%   8270.963 0.009560511 0.01454580
## 50%   15020.052 0.021118800 0.03062623
## 97.5% 30359.691 0.039317031 0.04732340
```

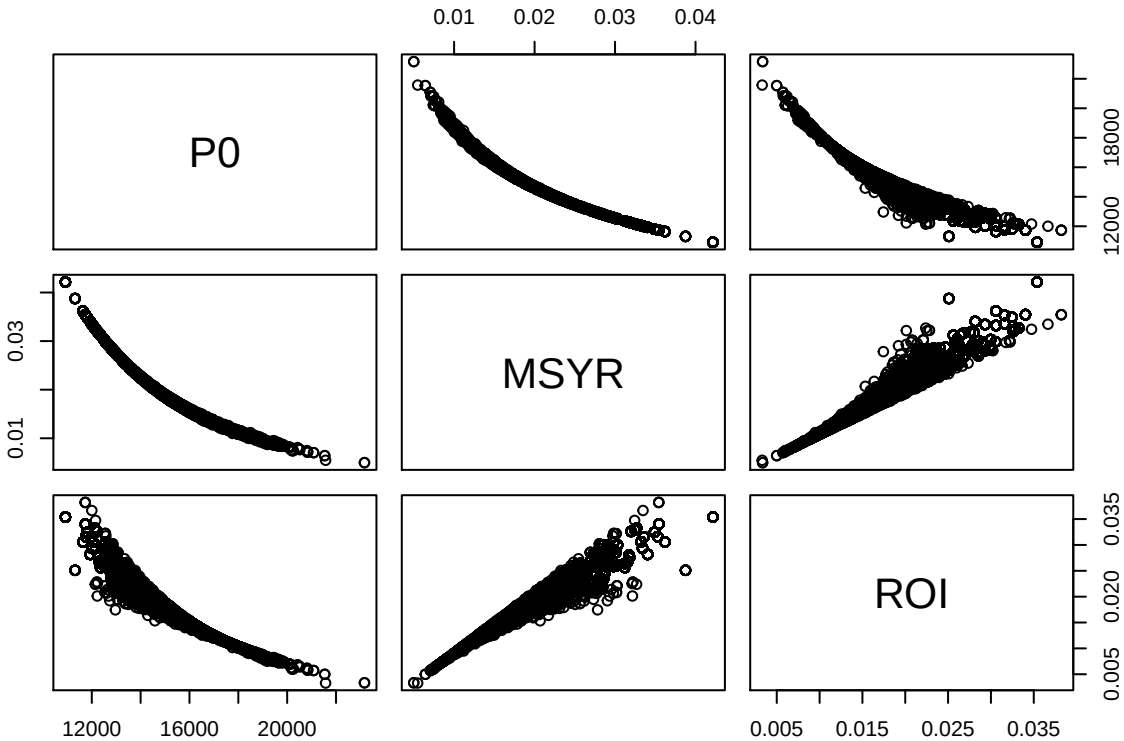
```r
apply(Theta.posterior, 2,
      function(x) c(mean = mean(x), quantile(x, probs = c(.025, .5, .975))))
```

```
##             P0       MSYR        ROI
## mean  14329.44 0.02220061 0.02087183
## 2.5%  11807.81 0.01170244 0.01075193
## 50%   14179.47 0.02177824 0.02051305
## 97.5% 17781.76 0.03491009 0.03158758
```

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

```
##              P0       MSYR        ROI
## P0    1.0000000 -0.9725900 -0.9661696
## MSYR -0.9725900  1.0000000  0.9747239
## ROI  -0.9661696  0.9747239  1.0000000
```

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



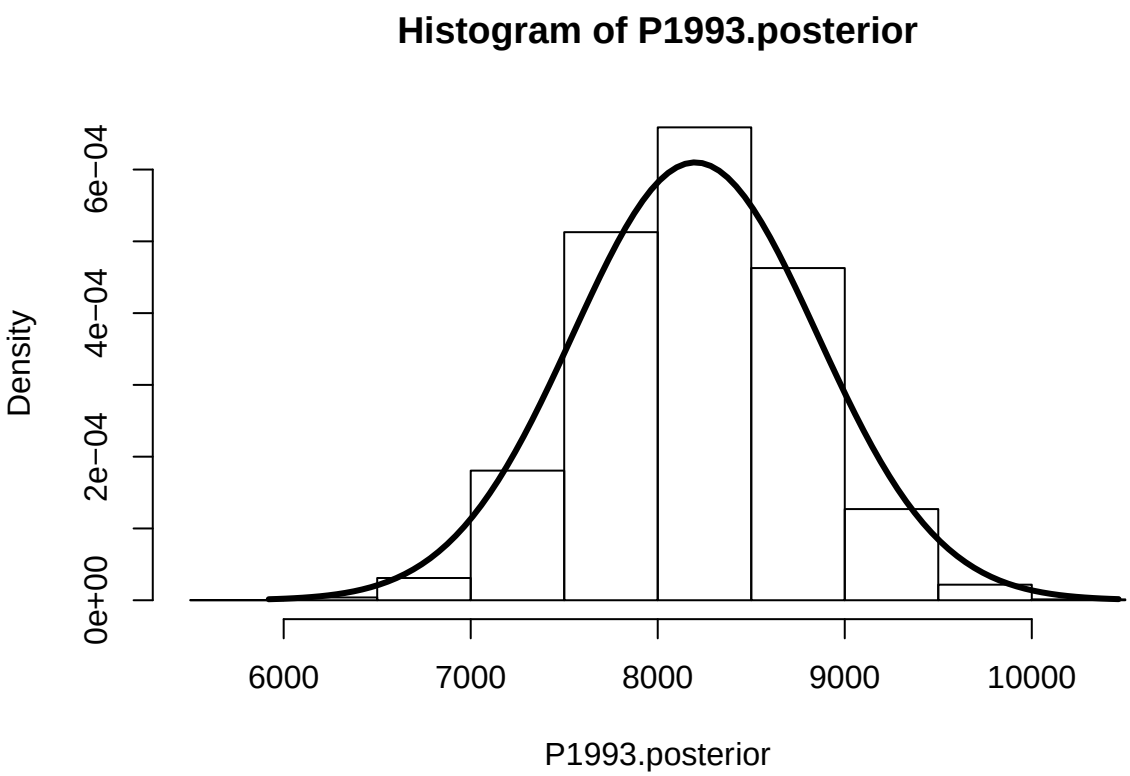

```r
# Now the output
P1993.posterior <- SpIR.posterior$phi.resamp
c(mean = mean(P1993.posterior), quantile(P1993.posterior, probs = c(.025, .5, .975)))
```

```
##     mean     2.5%      50%    97.5%
## 8185.642 7068.486 8181.697 9301.077
```

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

```
##      mean      2.5%       50%     97.5%
##  7799.024  5237.269  7797.065 10343.074
```

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

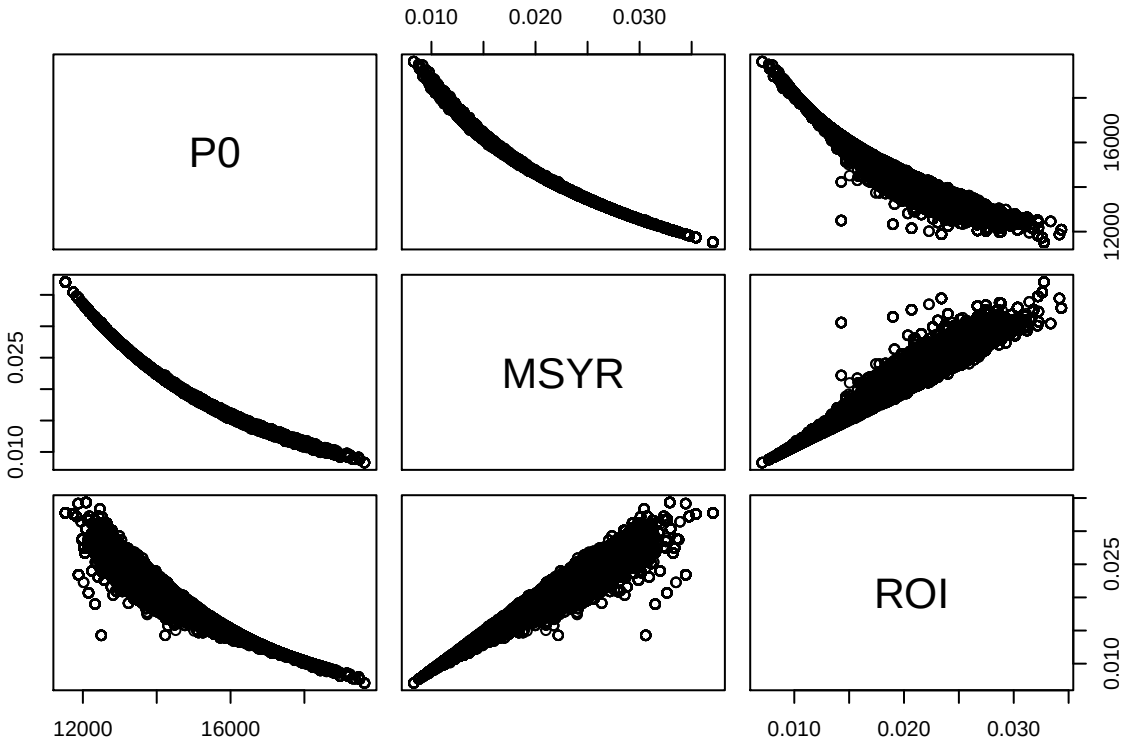



```
###########################
###########################
# Now let's approximate the posterior of theta using a different method
## Priors
prior <- function(param){
  dgamma(param[1] - 6400, shape = 2.8085, rate = 0.0002886, log = TRUE) +
    dgamma(param[2], shape = 8.2, rate = 372.7, log = TRUE)  +
    log(P1993.pooledPrior(getP1993(param[1:2]), weights = c(Alpha, 1-Alpha)))
}
## Likelihoods
L2 <- function(x){
  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, verbose = FALSE){
  Lk <- Lik(param)
  pr <-  prior(param)
  if(verbose) cat("Likelihood:", Lk, " prior:", pr,
                  " P0:", param[1], " MSYR:", param[2], "\n")
  return (Lk + pr)
}
##
# Let's run the stuff
```

```r
Niter <- 2 * M
run_metropolis_MCMC <- function(startvalue, Niter){
  require(MHadaptive)
  chain <- Metro_Hastings(li_func = posterior, pars = startvalue,
                          iterations = Niter, burn_in = round(.5 * Niter),
                          prop_sigma = var(Theta.posterior[, 1:2]),
                          par_names = c('P0', 'MSYR'),
                          quiet = FALSE)
  return(chain)
}
system.time(
  Chain <- run_metropolis_MCMC(
    startvalue = c(mean(P0.prior), mean(MSYR.prior)),
    Niter = Niter)
)
```

```
## Loading required package: MHadaptive

## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked _by_ '.GlobalEnv':
##
##     fbeta

## [1] "updating: 10%"
## [1] "updating: 20%"
## [1] "updating: 30%"
## [1] "updating: 40%"
## [1] "updating: 50%"
## [1] "updating: 60%"
## [1] "updating: 70%"
## [1] "updating: 80%"
## [1] "updating: 90%"
## [1] "updating: 100%"

##     user   system  elapsed
## 10935.43    14.74 10953.11
```

```r
MH.Post <- data.frame(Chain$trace,
                      ROI = apply(Chain$trace, 1, getROI))
names(MH.Post) <- c("P0", "MSYR", "ROI")
plot(MH.Post)
apply(MH.Post, 2, coda::effectiveSize)
```

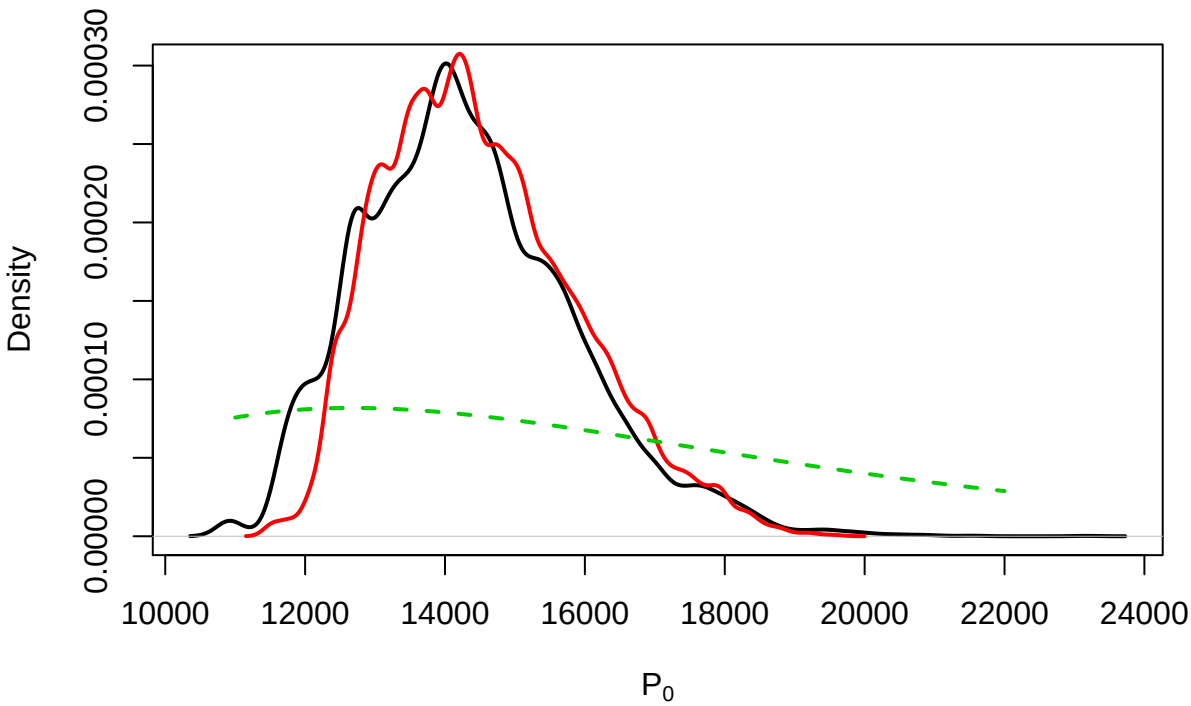


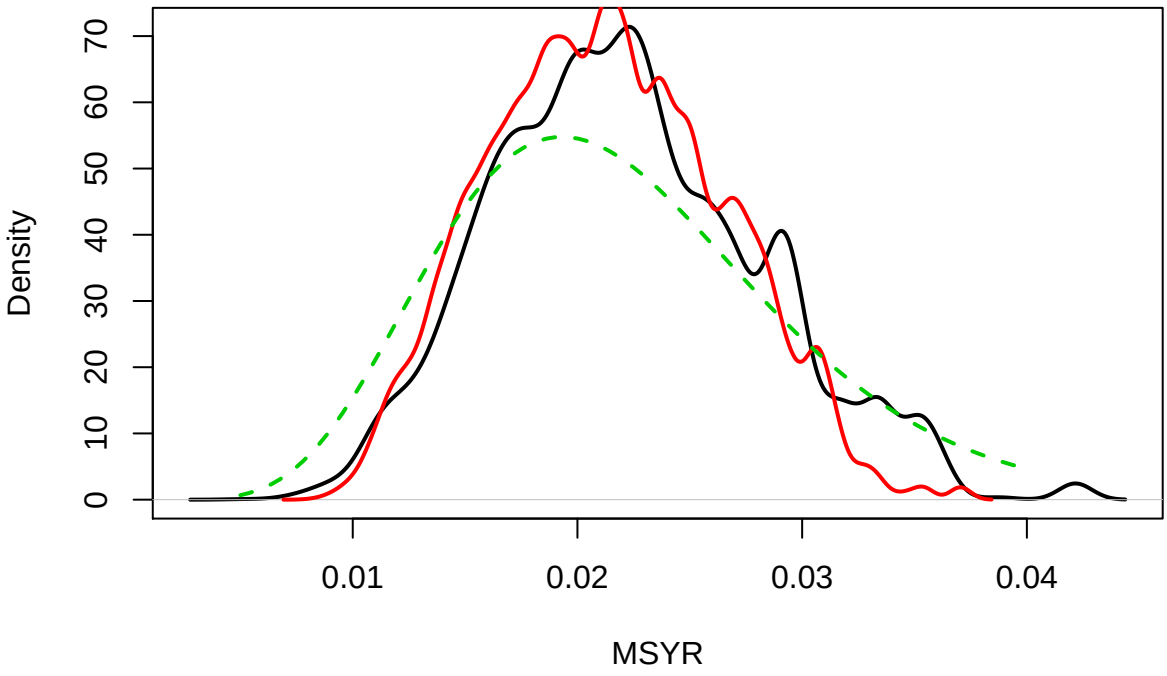

```
##       P0     MSYR      ROI
## 621.2899 657.7927 691.2110
```

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

```
##             P0       MSYR        ROI
## mean  14517.20 0.02127948 0.01998002
## 2.5%  12377.48 0.01201940 0.01115763
## 50%   14344.15 0.02114024 0.01994878
## 97.5% 17589.90 0.03101516 0.02970779
```

```r
# 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)
```




```
#
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)
```