

Simultaneous count model vignette

G.E. Ryan

2019-10-25

This is a vignette illustrating the "simultaneous count model" approach from *Simultaneous-count models to estimate abundance from counts of unmarked individuals with imperfect detection*, *Conservation Biology* 33(3) 697–708, G.E. Ryan et al. 2019.

Packages

```
library(dplyr)
library(rjags)
library(R2jags)
library(jagstools)
library(ggplot2)
library(bayesplot)
```

If necessary install `jagstools` from [github](#) with:

```
library(remotes)

remotes::install_github("johnbaums/jagstools")
```

Data

Create a dataset simulating population of 1000 individuals,

A survey effort at:

- 5 sites,
- visited 4 times/year,
- for 5 years

Around half of the population is ever observed at each site in a single sampling occasion (visit).

Probabilities of detection are constant at each site over time.

Simulated data

```
# population size
population.size <- 1000

# site probabilities of detection [1:5], and probability of non-detection [6]
probs <- c(0.01, 0.15, 0.04, 0.22, 0.08, 0.5)

dat <- array(
  data = rmultinom(
    n = 20, # 4 visits x 5 years
    size = population.size,
    prob = probs
  ),
  dim = c(6,4,5), # sites +1, visits/year, years
  dimnames = list(
    "site" = sprintf("site%s", 1:6),
    "visit" = sprintf("visit%s", 1:4),
    "year" = sprintf("year%s", 1:5)
  )
)
```

Let's look at the first two years of data:

```
print(dat[, ,1:2])

## , , year = year1
##      visit
## site  visit1 visit2 visit3 visit4
## site1    5    13    10    11
## site2   159   128   158   156
## site3    40    44    38    41
## site4   215   219   214   221
## site5    71    73    74    80
## site6   510   523   506   491
##
## , , year = year2
##      visit
## site  visit1 visit2 visit3 visit4
## site1    11    13    13    13
## site2   151   157   166   161
## site3    46    42    51    33
## site4   203   208   214   217
## site5    78    84    72    77
## site6   511   496   484   499
```

We can see that this simulates six sites, because while we visit five sites, this simulated sixth site represents the population that we have not observed,

Model data

To model the observation process, we won't observe the individuals at site six, so we record these in the array as `NA`. We'll use `y` for these data simulated observation data to be consistent with the notation in Ryan *et al.*

```
y <- dat

y[6,] <- NA
```

Let's have a look at the first two years of these data

```
print(y[, ,1:2])

## , , year = year1
##      visit
## site  visit1 visit2 visit3 visit4
## site1    5    13    10    11
## site2   159   128   158   156
## site3    40    44    38    41
## site4   215   219   214   221
## site5    71    73    74    80
## site6    NA    NA    NA    NA
##
## , , year = year2
##      visit
## site  visit1 visit2 visit3 visit4
## site1    11    13    13    13
## site2   151   157   166   161
## site3    46    42    51    33
## site4   203   208   214   217
## site5    78    84    72    77
## site6    NA    NA    NA    NA
```

Modelling

Write our JAGS model.

This model For explanation of ones trick see: <https://www.mrc-bsu.cam.ac.uk/wp-content/uploads/manual14.pdf>

```
cat('
data{
  for (j in 1:nvisits){
    for (k in 1:nyears){
      ones[j,k] <- 1
      # constants for ones trick
    }
  }
  C <- 10000
  # constant for ones trick
}

model{
  lambda ~ dgamma(1e-5, 1e-5)T(1,5000)
  # Hyper-prior for population size, truncated at 5000 to improve convergence

  N ~ dpois(lambda)
  # Prior estimate of population size

  pi[1:(nsites + 1)] ~ ddirich(alpha[1:(nsites + 1)])
  # Prior for our observation process

  for (k in 1:nyears){
    for (j in 1:nvisits){
      seen[j,k] <- sum(y[1:nsites,j,k]) # Number of individuals observed
      y[(nsites + 1),j,k] <- N - seen[j,k] # complete our array y

      # Likelihood written in full
      # because we cant simply do y ~ multinomial with missing data in JAGS
      for (i in 1:(nsites + 1)){
        lp[i,j,k] <- y[i,j,k]*log(pi[i]) - logfact(y[i,j,k])
      }
      ln[j,k] <- sum(lp[1:(nsites + 1),j,k]) + logfact(N)
      log(P[j,k]) <- ln[j,k] - log(C)
      ones[j,k] ~ dbern(P[j,k]) # ones trick
    }
  }
}
', file = (jags.mod <- tempfile())) # This writes a temporary file with the model and assigns the path at jags.mod
```

Prepare data for JAGS

```
nsites <- dim(y)[1] - 1 # should be 5
nvisits <- dim(y)[2] # should be 4
nyears <- dim(y)[3] # should be 5

alpha <- rep(1, (nsites + 1))

jags.params <- c("N", "pi", "lambda")

jags.inits <- function(){
  list(
    lambda = 2*sum(y[,1,1], na.rm = TRUE) # initial value to search from is 5x the total observed in the first sampling occasion
  )
}

jags.data <- list(
  "y",
  "nsites",
  "nvisits",
  "nyears",
  "alpha"
)

assign("y", y, envir=globalenv())
assign("alpha", alpha, envir=globalenv())
assign("nsites", nsites, envir=globalenv())
assign("nvisits", nvisits, envir=globalenv())
assign("nyears", nyears, envir=globalenv())
```

Fit our model

This is a short-run test fit, with just 5000 iterations.

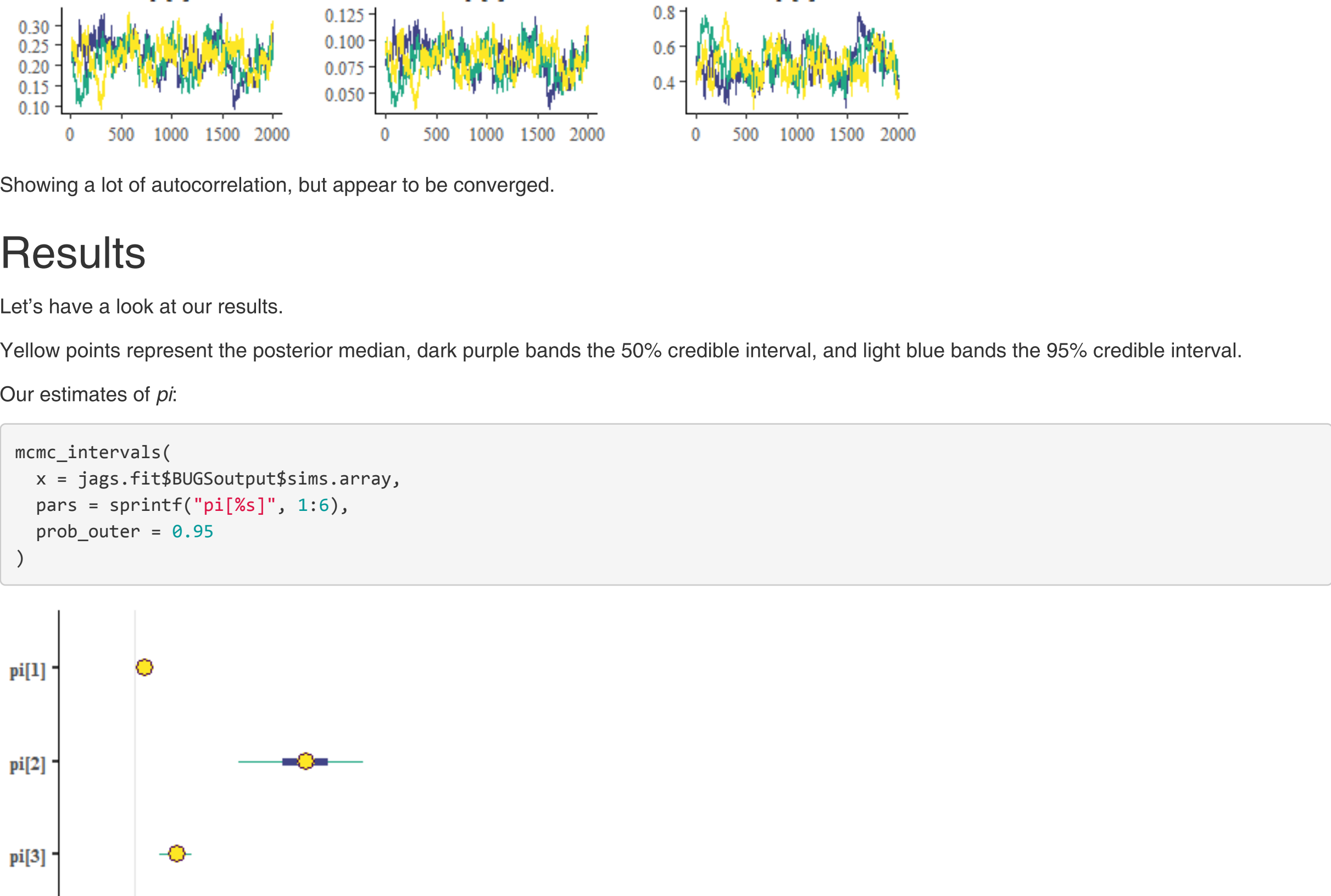
```
Inference for Bugs model at "D:\Users\ryan\AppData\Local\Temp\RtmpwX8kR1\filefa075035089", fit using jags,
3 chains, each with 5000 iterations (first 100 discarded), n.thin = 4
n.sims = 3675 iterations saved
      mu.vect sd.vect   2.5%   25%   50%   75%   97.5% Rhat n.eff
N      623.108  22.414 590.000 602.000 626.000 636.000 682.000 1.520   8
lambda 623.438 33.556 560.754 600.022 622.965 645.199 691.448 1.196  15
pi[1]   0.016  0.001  0.014  0.015  0.016  0.017  0.018 1.059  130
pi[2]   0.241  0.010  0.223  0.237  0.242  0.250  0.259 1.661   6
pi[3]   0.062  0.003  0.055  0.061  0.062  0.063  0.066 1.207  19
pi[4]   0.353  0.013  0.321  0.345  0.349  0.364  0.378 1.454   8
pi[5]   0.131  0.006  0.118  0.127  0.130  0.137  0.140 1.411   9
pi[6]   0.197  0.029  0.153  0.167  0.202  0.209  0.267 1.587   7
deviance 1075.988  7.957 1063.426 1070.355 1074.338 1081.477 1090.652 1.317  10

For each parameter, n.eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

DIC info (using the rule, pD = var(deviance)/2)
pD = 24.8 and DIC = 1100.8
DIC is an estimate of expected predictive error (lower deviance is better).
```

The high R-hat values suggest the model is not converged.

Let's also check the traceplots.



These are clearly not converged.

Let's try fitting with a much longer run and burn-in. Generally this might be done iteratively, but let's go straight to 300k iterations with 100k burn-in.

We also use `jags.parallel` here to fit chains in parallel.

```
jags.fit <- do.call(
  what = jags.parallel,
  args = list(
    data = jags.data,
    inits = jags.inits,
    parameters.to.save = jags.params,
    n.iter = 300000,
    n.burnin = 100000,
    n.chains = 3,
    n.thin = 100,
    model.file = jags.mod
  )
)

Inference for Bugs model at "D:\Users\ryan\AppData\Local\Temp\RtmpwX8kR1\filefa075035089", fit using jags,
3 chains, each with 3e+05 iterations (first 1e+05 discarded), n.thin = 100
n.sims = 6000 iterations saved
      mu.vect sd.vect   2.5%   25%   50%   75%   97.5% Rhat n.eff
N      1046.889 228.312 752.975 887.000 997.000 1153.000 1661.050 1.012  200
lambda 1047.262 230.544 746.132 886.918 997.805 1157.047 1659.316 1.011  200
pi[1]   0.010  0.002  0.006  0.008  0.010  0.011  0.013 1.011  200
pi[2]   0.150  0.028  0.091  0.131  0.152  0.170  0.202 1.011  200
pi[3]   0.037  0.007  0.023  0.033  0.038  0.043  0.050 1.011  200
pi[4]   0.218  0.040  0.132  0.190  0.220  0.247  0.291 1.011  200
pi[5]   0.081  0.015  0.048  0.070  0.081  0.091  0.108 1.012  200
pi[6]   0.504  0.092  0.337  0.437  0.500  0.568  0.700 1.012  180
deviance 1053.474  3.470 1048.578 1050.933 1052.834 1055.376 1061.920 1.002  2200

For each parameter, n.eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

DIC info (using the rule, pD = var(deviance)/2)
pD = 6.0 and DIC = 1059.5
DIC is an estimate of expected predictive error (lower deviance is better).
```

These R-hat values are much better (<1.1).

Let's also see some traceplots.



Showing a lot of autocorrelation, but appear to be converged.

Results

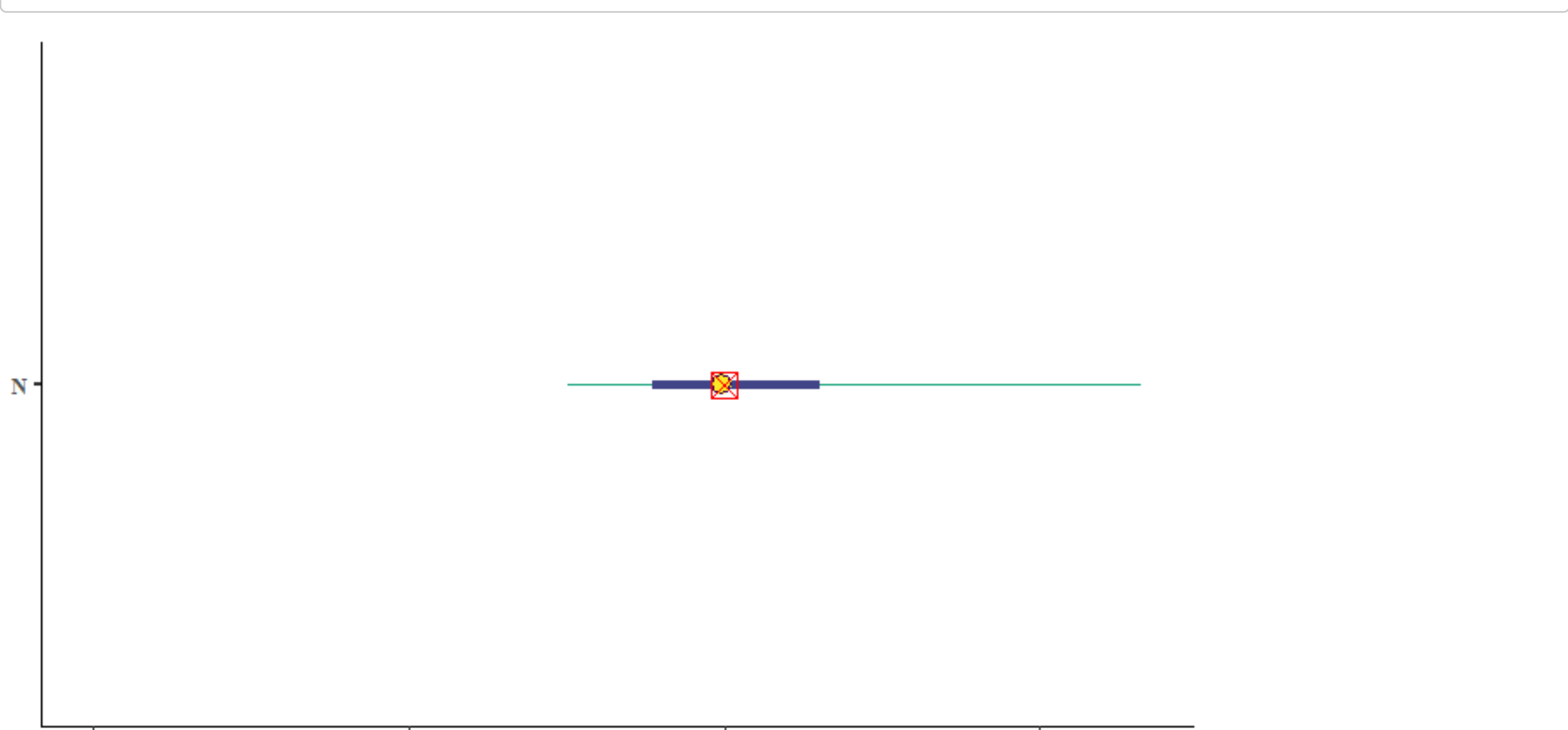
Let's have a look at our results.

Yellow points represent the posterior median, dark purple bands the 50% credible interval, and light blue bands the 95% credible interval.

Our estimates of π :



And of N :



Scale for 'x' is already present. Adding another scale for 'x', which will replace the existing scale.

Looks pretty good!

And of N :

Scale for 'x' is already present. Adding another scale for 'x', which will replace the existing scale.

Not too bad.