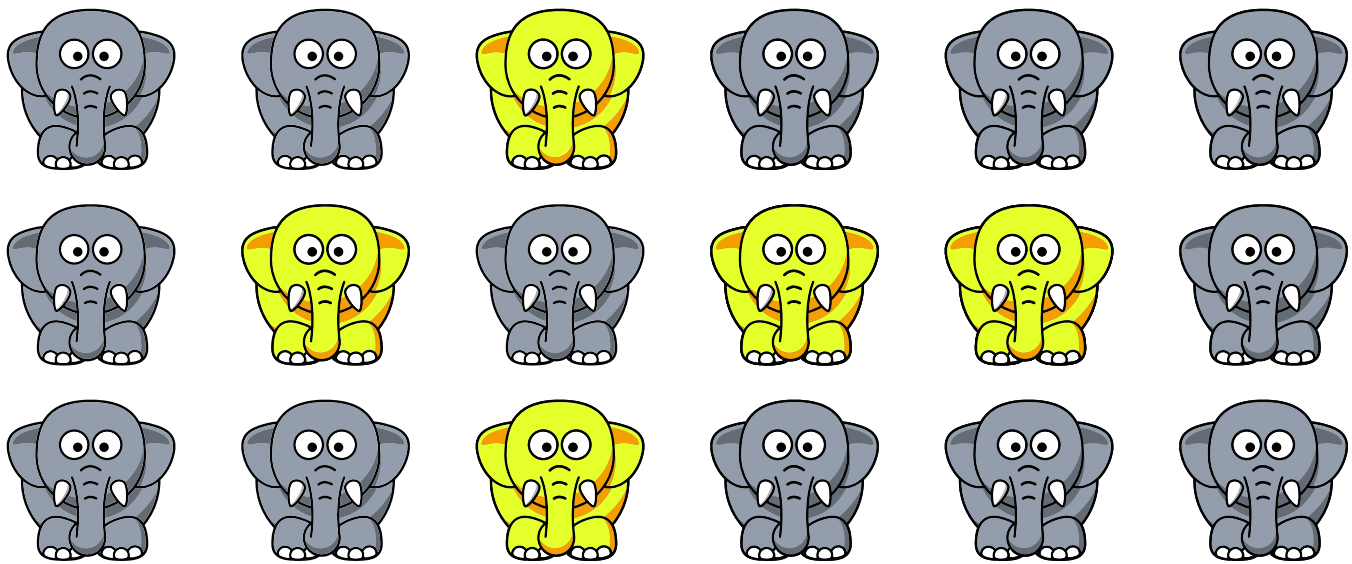
BISC-869, Mark-Recapture & Occupancy

April 8, 2020

Let's think back to the logic of our elephant population size estimation example from workshop.

Suppose we go out and catch 5 elephants and mark them.

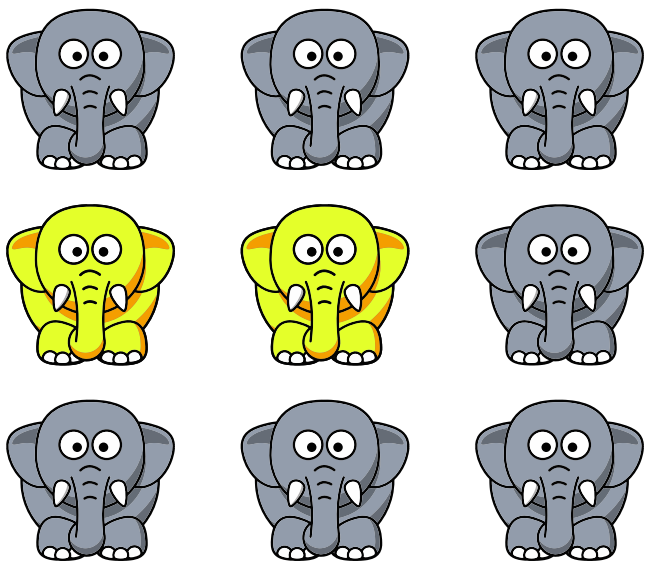
These elephants represent some unknown fraction of the total population (in this case, there are 18 in total).



Now, suppose we go back and capture 9 elephants.

If every elephant has an equal chance of being caught, the number of elephants that are marked versus unmarked in this sample should tell us something about how many elephants there are in total.

We used a hypergeometric distribution which calculates the probabilility of drawing m marked individuals (2 in this example) in a sample of n individuals (9 in this example) if there are a total of M marked (5 here) and U unmarked individuals (13 here, but this is the quantity we'd be estimating).



What happens if some elephants are more likely to get caught than others (e.g., larger ones might be easier to see)?

And what if we want to estimate more than just the population size? What if we want to estimate survival and ask how survival depends on attributes of an elephant's location or phenotype?

To answer questions like this, we have to use a more complex framework that allows us to model both the *recapture* process and individual's *rates of survival*, for example.

But, all we can do is go out and recapture individuals. Thus, our model must estimate both an individual's probability of being captured **AND** its probability of being alive. If it is not alive, it cannot be recaptured. If it is captured, it must still be alive...

This is a **Hierarchical Model**. There are numerous powerful and new frameworks for such modelss. For example, `WinBUGS` or `JAGS` and, more recently, `Nimble` and `Stan`.

Gibbs sampling is a method for sampling from distributions over at least two dimensions.

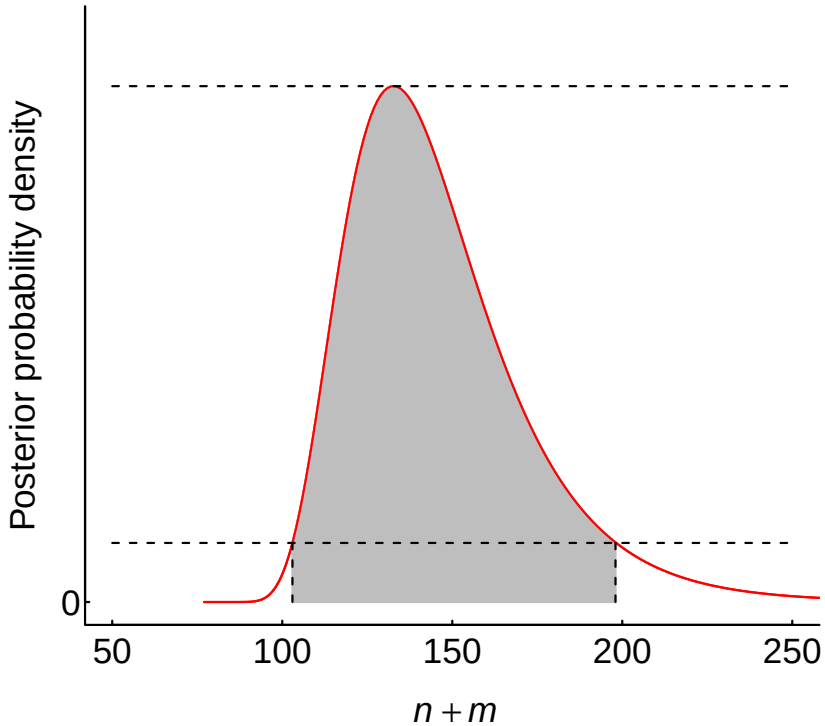
It is a Markov chain Monte Carlo (MCMC) algorithm.

The power of JAGS is that you can build models yourself without worrying too much about the analytical tractability of the likelihood (we will discuss "convergence" later).

This example continues one from the likelihood and Bayesian workshops. Eggert et al. (2003.*Molecular Ecology* 12: 1389-1402) used mark-recapture methods to estimate the total number of forest elephants inhabiting Kakum National Park in Ghana by sampling dung and extracting elephant DNA to uniquely identify individuals.

Over the first seven days of collecting the researchers identified 27 elephant individuals. We will refer to these 27 elephants as *marked*. Over the next eight days, they sampled 74 individuals, of which 15 had been previously marked. We refer to these 15 elephants as *recaptured*.

We used first principles to construct a posterior distribution and 95% BCI.



We found that the maximum posterior probability was 133 elephants and the 95% BCI spanned the interval [103,199].

Now let's do the same thing in JAGS.

First, construct a list containing all data that will be passed in to JAGS.

```
my.data <- list(n.marked=27, n.capture=74, n.recapture=15,
n.newcapture=74-15)
```

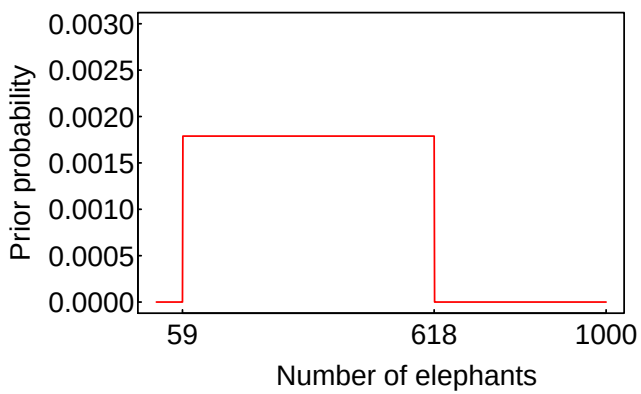
```
$n.marked
[1] 27

$n.capture
[1] 74

$n.recapture
[1] 15

$n.newcapture
[1] 59
```

Next, construct the model

```
my.model <- function() {
  ## priors
  for(i in 1:(n.newcapture-1)) {
    prob[i] <- 0
  }
  for(i in n.newcapture:(n.newcapture+500)) {
    prob[i] <- 1
  }
  n.unmarked ~ dcat(prob)
  ## likelihood
  n.recapture ~ dhyper(n.marked, n.unmarked, n.capture, 1)
  ## parameter of interest
  n.total <- n.marked + n.unmarked
}
```

The categorical prior we have constructed is a uniform distribution over the specified range of possible total counts of unmarked elephants.

We chose 500 arbitrarily. If it turns out to be too small, we can increase and re-run the model.



Next, construct the model

```
my.model <- function() {
  ## priors
  for(i in 1:(n.newcapture-1)) {
    prob[i] <- 0
  }
  for(i in n.newcapture:(n.newcapture+500)) {
    prob[i] <- 1
  }
  n.unmarked ~ dcat(prob)
  ## likelihood
  n.recapture ~ dhyper(n.marked, n.unmarked, n.capture, 1)
  ## parameter of interest
  n.total <- n.marked + n.unmarked
}
```

JAGS also requires a function that returns a list of "inits". These are starting values for the MCMC chains. I'm going to give it an empty list, which means it will draw initial values from the priors.

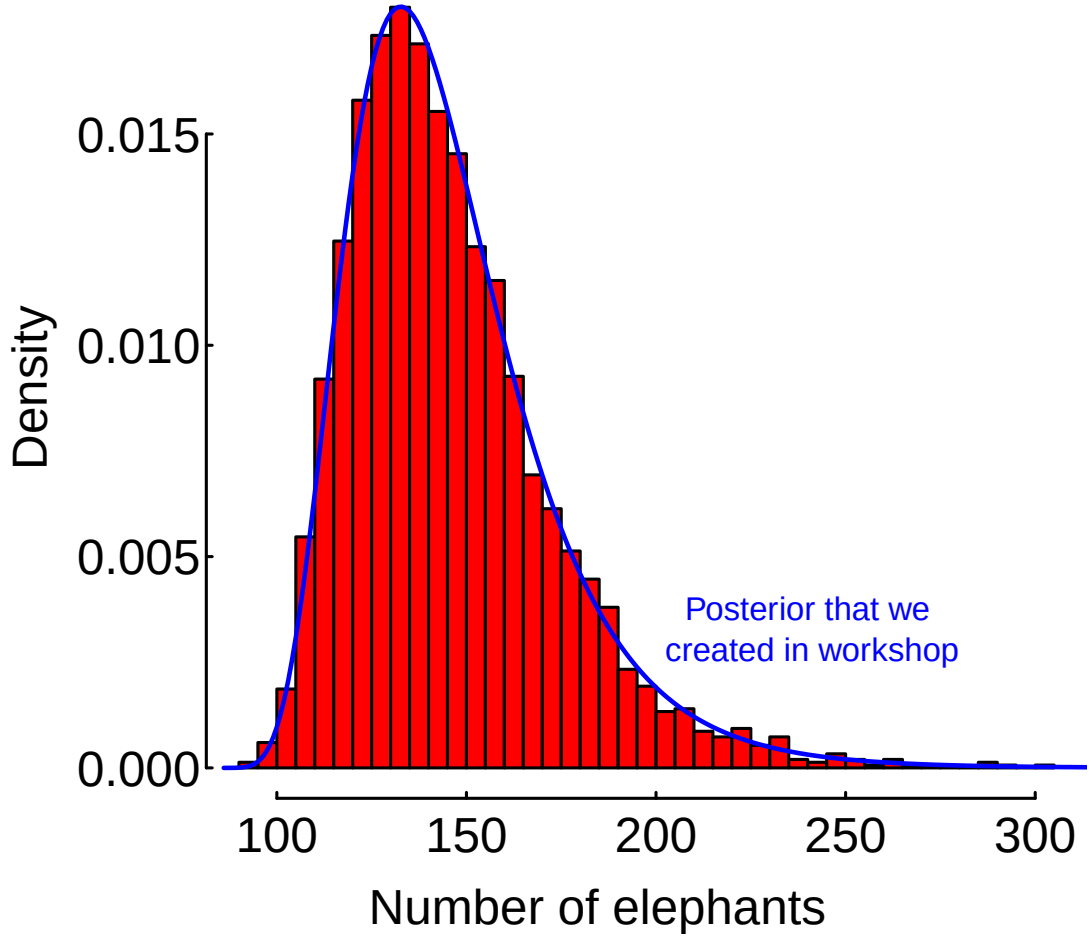
```
my.inits <- function() {
  list()
}
```

Finally, we can run the model

```
res <- jags(data=my.data,
            inits=my.inits,
            parameters.to.save=c('n.total', 'n.unmarked'),
            model.file=my.model,
            n.iter=11000,
            n.burnin=1000,
            n.thin=10,
            n.chains=3,
            working.directory=NULL)
```

In workshop this week, we will go over the basics of how to extract and examine the ouptut.

Posterior distribution for total number of elephants.



Extract maximum posterior probability:

```
tab <- table(res$BUGSoutput$sims.array[,,'n.total'])
tab[which.max(tab)]
```

which yields 135 as our estimate (close to the 133 we calculated in workshop).

We can also calculate a 95% highest probability density region (one for each of our chains). This is equivalent to the 95% BCI we calculated before. Use the `coda` libary:

```
res.out <- coda.samples(res$model, c('n.total'), n.iter=11000)
HPDinterval(res.out)
```

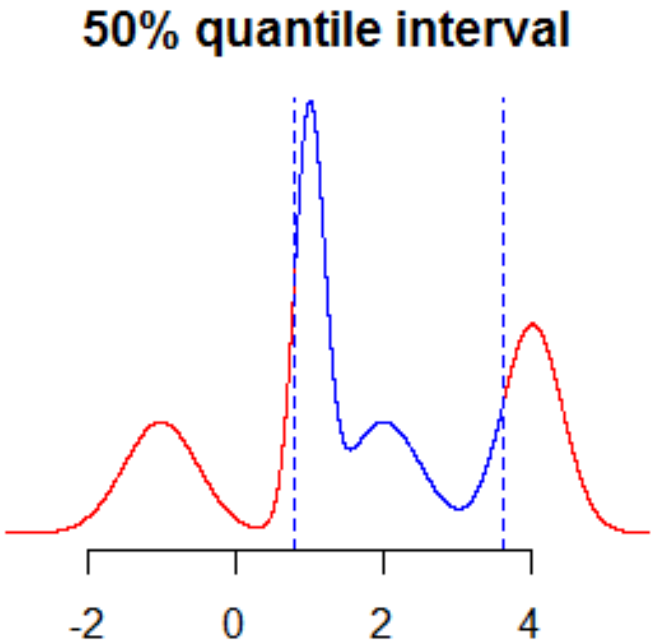
```
[[1]]
        lower upper
n.total   104   199
attr(,"Probability")
[1] 0.95

[[2]]
        lower upper
n.total   100   197
attr(,"Probability")
[1] 0.95

[[3]]
        lower upper
n.total   103   199
attr(,"Probability")
[1] 0.95
```

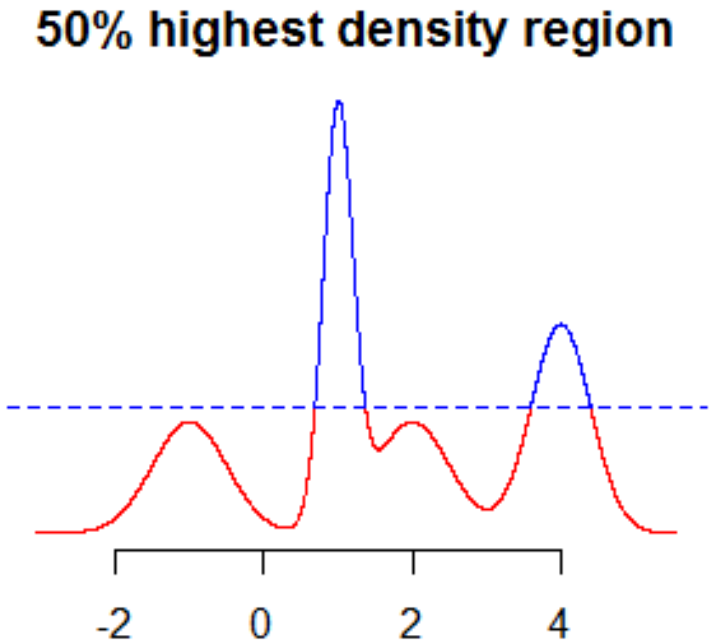
Compare to the BCI we calculated in workshop → [103,199].

```
round(res$BUGSoutput$summary, digits=1)
```

```
            mean   sd  2.5%   25%   50%   75% 97.5% Rhat n.eff
deviance     4.7  1.6   3.5   3.6   4.1   5.1   9.4    1  2400
n.total    145.9 26.8 108.0 127.0 141.0 160.0 213.0    1  3000
n.unmarked 118.9 26.8  81.0 100.0 114.0 133.0 186.0    1  3000
```

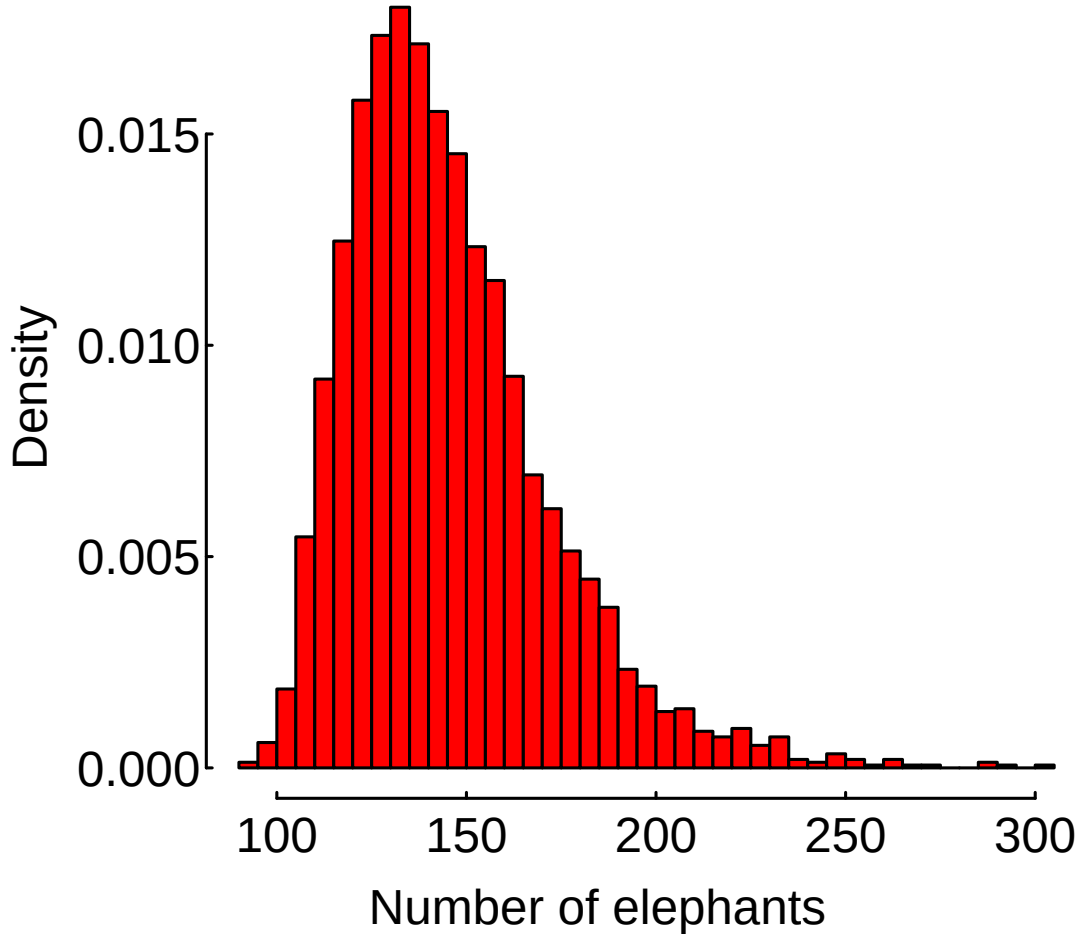
Why do these numbers not match our BCI?



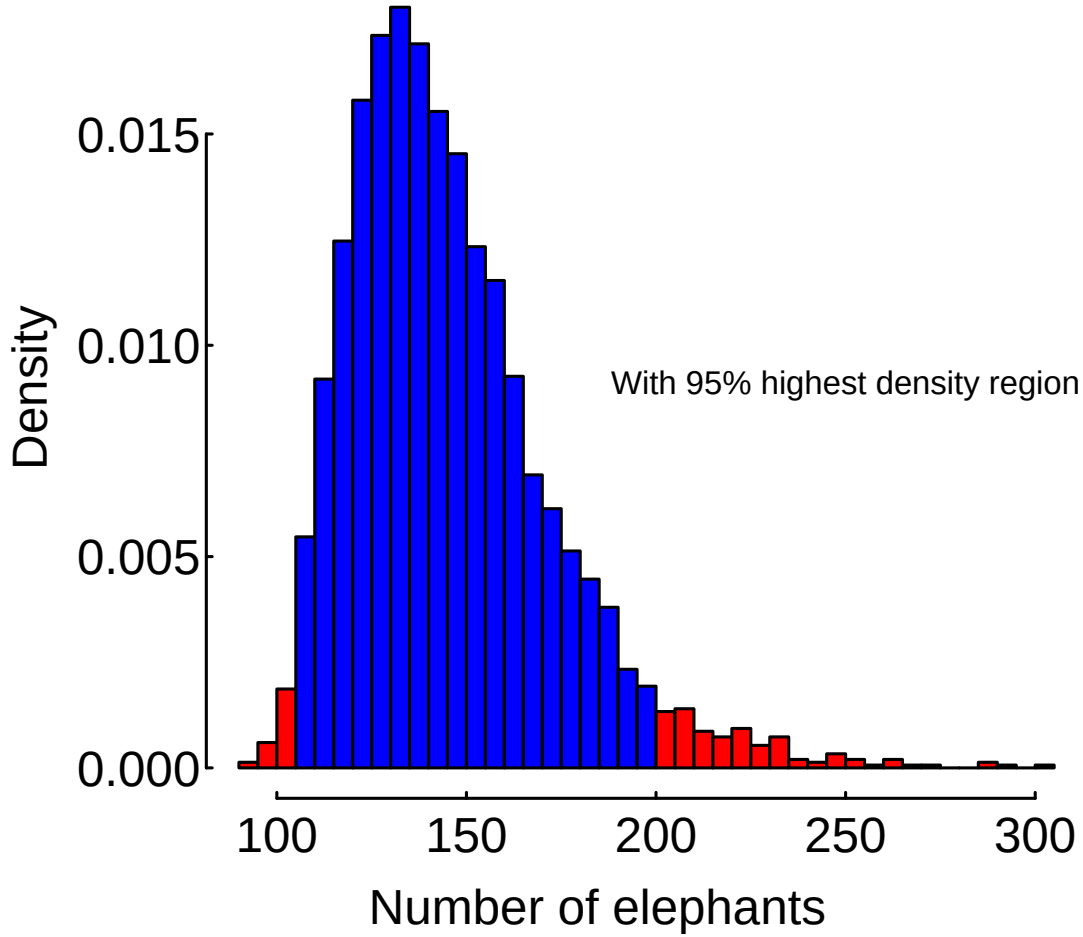


For unimodal, symmetric distributions, these will be the same. Otherwise, maybe not.

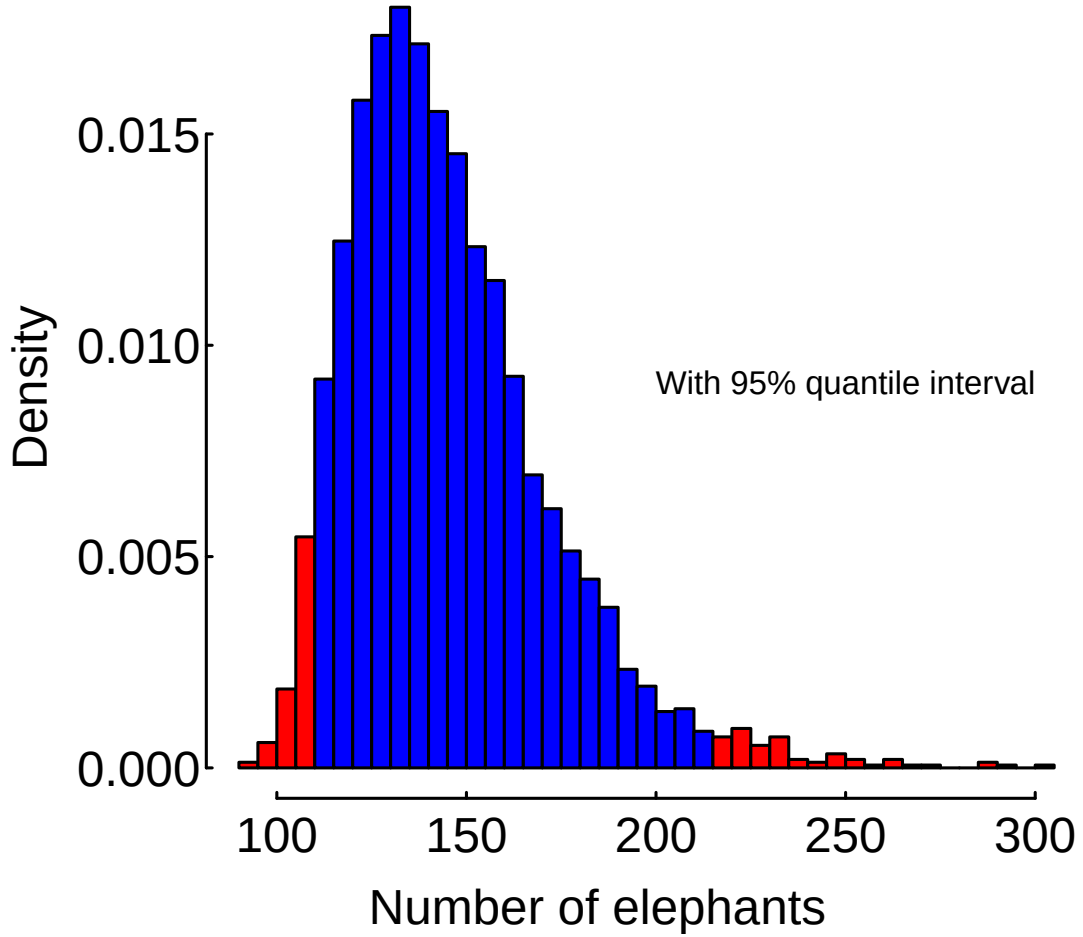
Posterior distribution for total number of elephants.



Posterior distribution for total number of elephants.



Posterior distribution for total number of elephants.

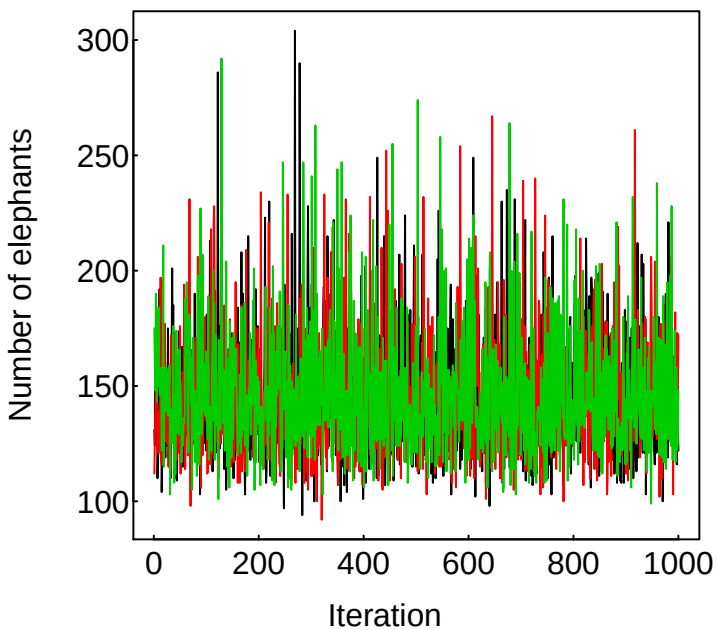


Assessing convergence

`Rhat` in your summary table should be near 1 for every parameter (here we have values extremely close to 1).

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
deviance	4.7	1.6	3.5	3.6	4.1	5.1	9.4	1	2400
n.total	145.9	26.8	108.0	127.0	141.0	160.0	213.0	1	3000
n.unmarked	118.9	26.8	81.0	100.0	114.0	133.0	186.0	1	3000

Your MCMC chains should look "grassy".

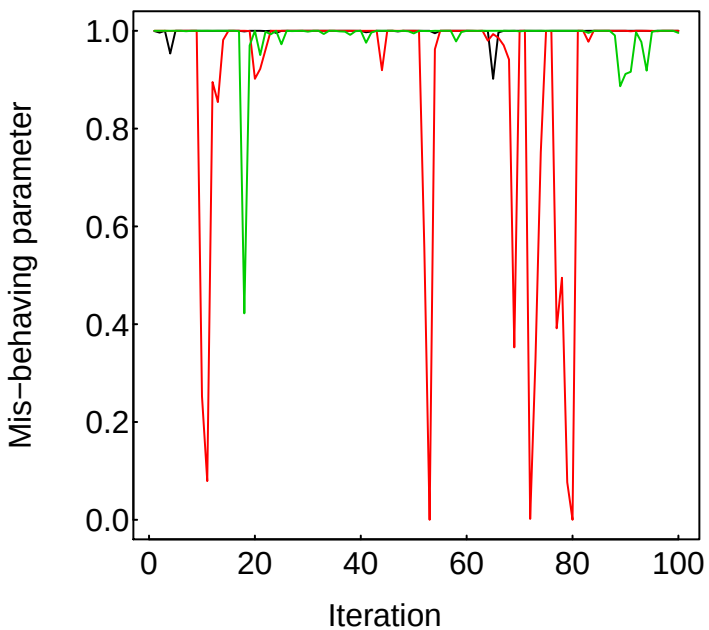


Assessing convergence

`Rhat` in your summary table should be near 1 for every parameter (here we have values extremely close to 1).

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
deviance	4.7	1.6	3.5	3.6	4.1	5.1	9.4	1	2400
n.total	145.9	26.8	108.0	127.0	141.0	160.0	213.0	1	3000
n.unmarked	118.9	26.8	81.0	100.0	114.0	133.0	186.0	1	3000

Bad chains take many forms. Here are some of mine.

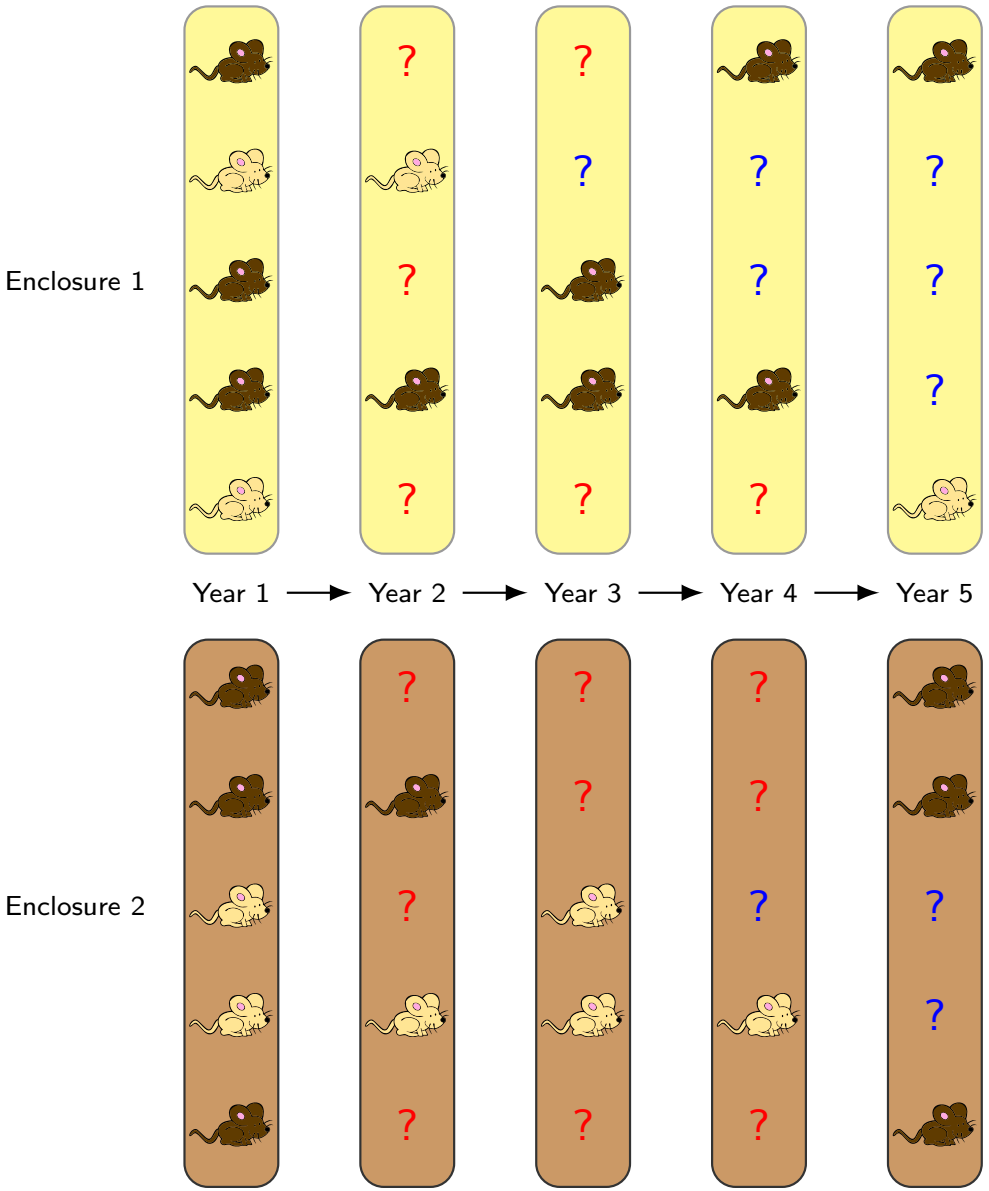


JAGS is simply the tool that carries out the MCMC algorithm to construct your chains. There are lots of others out there (e.g., NIMBLE and Stan are two new ones), and the logic applied here should generally apply to those as well.

Data: In 2011, Rowan Barrett and colleagues constructed field enclosures in Nebraska on both light and dark soils. They filled these with wild caught mice, collected from light and dark soils, and then recaptured mice each year to test whether survival was higher for mice that matched the enclosure soil type.

We will examine this data further in workshop.

Example: Cormack-Jolly-Seber Model



? ← We know these are missed captures.

? ← These could be missed captures, or the mouse might have died.

Based on the patterns of capture, it looks like mice that match the soil type live longer, but are also harder to detect.

In order to check this, we want to model both recapture rate and probability of survival as a function of mouse phenotype, soil type, and an interaction between the two.

What would such a model look like?

First, make a list containing data for JAGS:

```
my.data <- list(captured=captured,
                alive=alive,
                phenotype=as.integer(phenotype)-1,
                soil=as.integer(soil)-1,
                nind=nrow(captured),
                nrep=ncol(captured))
```

The data we pass in to JAGS is of two types:

1. The raw data

```
> head(my.data$captured)
      recap0 recap1 recap2 recap3 recap4 recap5
RB192      1      1      0      0      0      0
RB70       1      1      0      0      0      0
RB281      1      0      0      0      0      0
RB223      1      1      0      0      0      0
RB227      1      1      0      0      0      0
RB43       1      0      0      1      1      1
> head(my.data$phenotype)
[1] 0 1 0 1 1 0
> head(my.data$soil)
[1] 0 0 1 0 0 0
```

2. Some quantities we calculated

```
> head(my.data$alive)
      recap0 recap1 recap2 recap3 recap4 recap5
RB192      1      1     NA     NA     NA     NA
RB70       1      1     NA     NA     NA     NA
RB281      1     NA     NA     NA     NA     NA
RB223      1      1     NA     NA     NA     NA
RB227      1      1     NA     NA     NA     NA
RB43       1      1      1      1      1      1
> my.data$nind
[1] 297
> my.data$nrep
[1] 6
```

Note: for now, we are ignoring the random effect of 'enclosure'.

Example: Cormack-Jolly-Seber Model

The model:

```
my.model <- function() {

  ## priors (recapture)
  p.0                 ~ dnorm(0,0.001)
  p.soil              ~ dnorm(0,0.001)
  p.phenotype         ~ dnorm(0,0.001)
  p.soil.phenotype    ~ dnorm(0,0.001)
  ## priors (survival)
  phi.0               ~ dnorm(0,0.001)
  phi.soil            ~ dnorm(0,0.001)
  phi.phenotype       ~ dnorm(0,0.001)
  phi.soil.phenotype  ~ dnorm(0,0.001)

  ## for each individual
  for(ind in 1:nind) {

    ## model for recapture
    logit(p[ind]) <-
      p.0 +
      p.phenotype*phenotype[ind] +
      p.soil*soil[ind] +
      p.soil.phenotype*soil[ind]*phenotype[ind]

    ## model for survival
    logit(phi[ind]) <-
      phi.0 +
      phi.phenotype*phenotype[ind] +
      phi.soil*soil[ind] +
      phi.soil.phenotype*soil[ind]*phenotype[ind]

    ## likelihood
    for(rep in 2:nrep) {

      ## state equation
      alivep[ind,rep] <- phi[ind] * alive[ind,rep-1]
      alive[ind,rep] ~ dbern(alivep[ind,rep])

      ## observation equation
      sightp[ind,rep] <- p[ind] * alive[ind, rep]
      captured[ind,rep] ~ dbern(sightp[ind,rep])

    }
  }
}
```

Flat priors (JAGS uses inverse variance)

These should look familiar.
They are simply linear models with logit link.
Logistic regression, data is 0/1!

Loop starts at 2, because mice are all "alive" in first time step

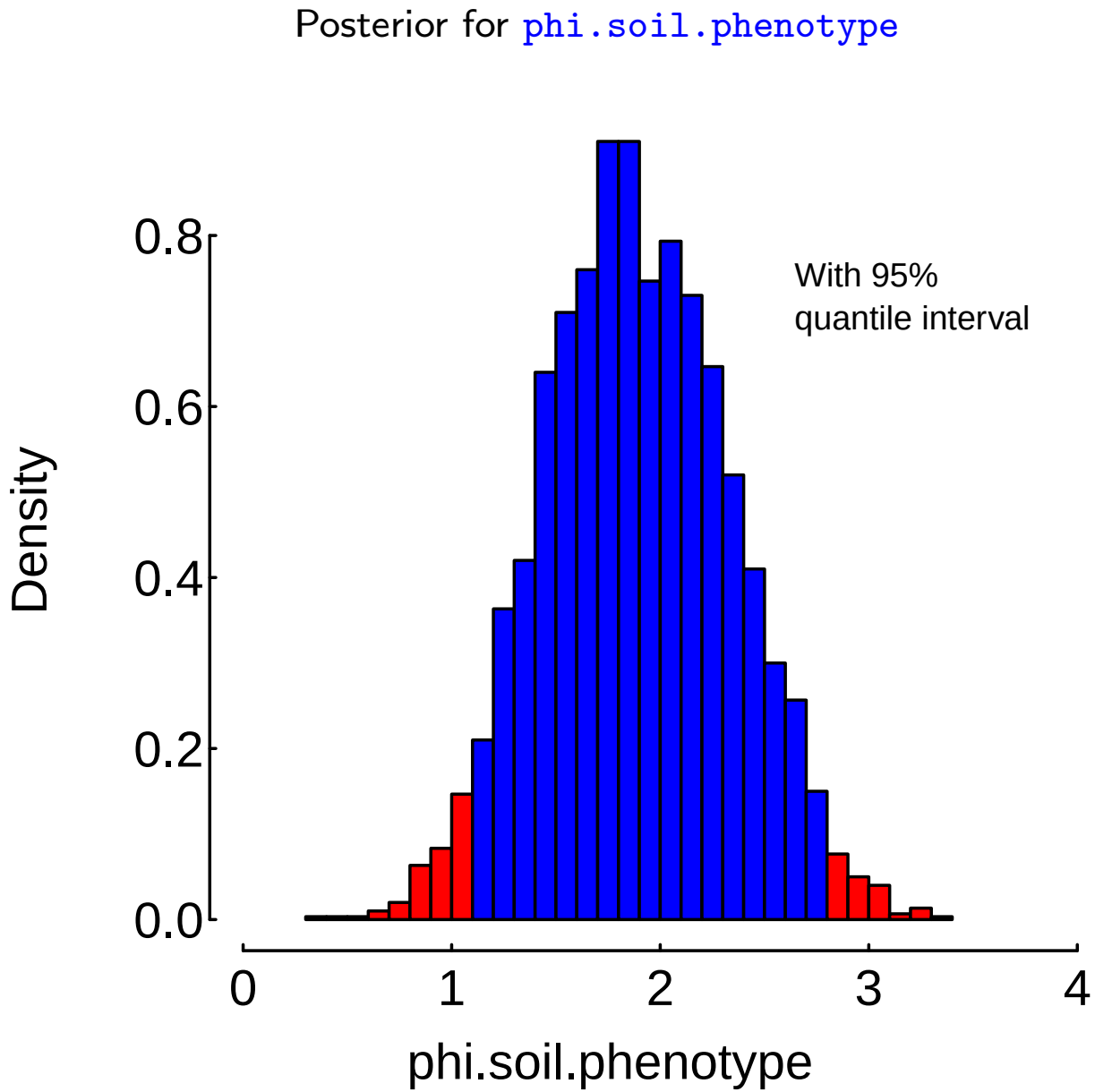
Example: Cormack-Jolly-Seber Model

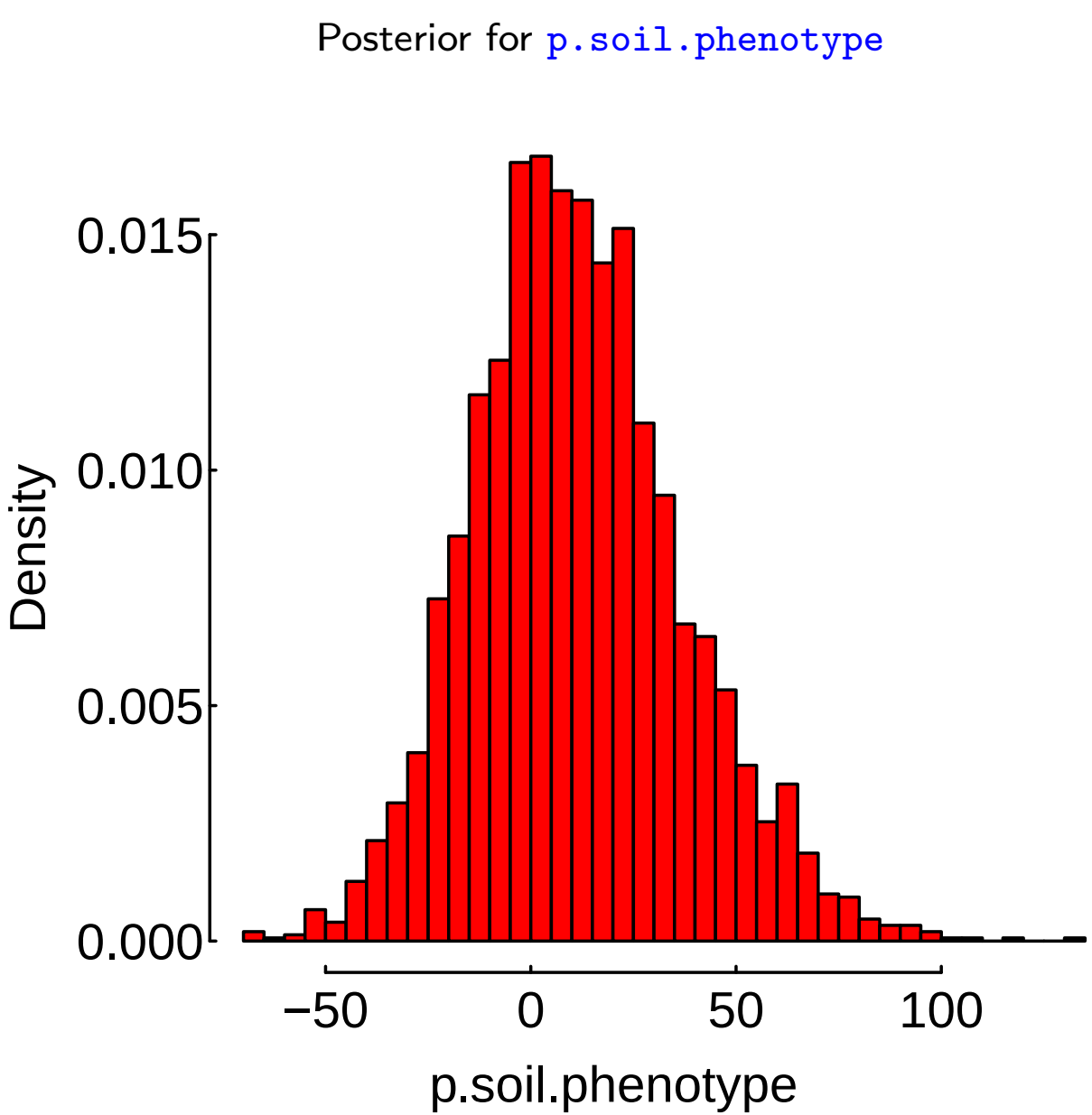
The model:

```
my.model <- function() {

  ## priors (recapture)
  p.0                ~ dnorm(0,0.001)
  p.soil             ~ dnorm(0,0.001)
  p.phenotype        ~ dnorm(0,0.001)
  p.soil.phenotype   ~ dnorm(0,0.001)
  ## priors (survival)
  phi.0              ~ dnorm(0,0.001)
  phi.soil           ~ dnorm(0,0.001)
  phi.phenotype      ~ dnorm(0,0.001)
  phi.soil.phenotype ~ dnorm(0,0.001)

  ## for each individual
  for(ind in 1:nind) {

    ## model for recapture
    logit(p[ind]) <-
      p.0 +
      p.phenotype*phenotype[ind] +
      p.soil*soil[ind] +
      p.soil.phenotype*soil[ind]*phenotype[ind]

    ## model for survival
    logit(phi[ind]) <-
      phi.0 +
      phi.phenotype*phenotype[ind] +
      phi.soil*soil[ind] +
      phi.soil.phenotype*soil[ind]*phenotype[ind]

    ## likelihood
    for(rep in 2:nrep) {

      ## state equation
      alivep[ind,rep] <- phi[ind] * alive[ind,rep-1]
      alive[ind,rep] ~ dbern(alivep[ind,rep])

      ## observation equation
      sightp[ind,rep] <- p[ind] * alive[ind, rep]
      captured[ind,rep] ~ dbern(sightp[ind,rep])

    }
  }
}
```

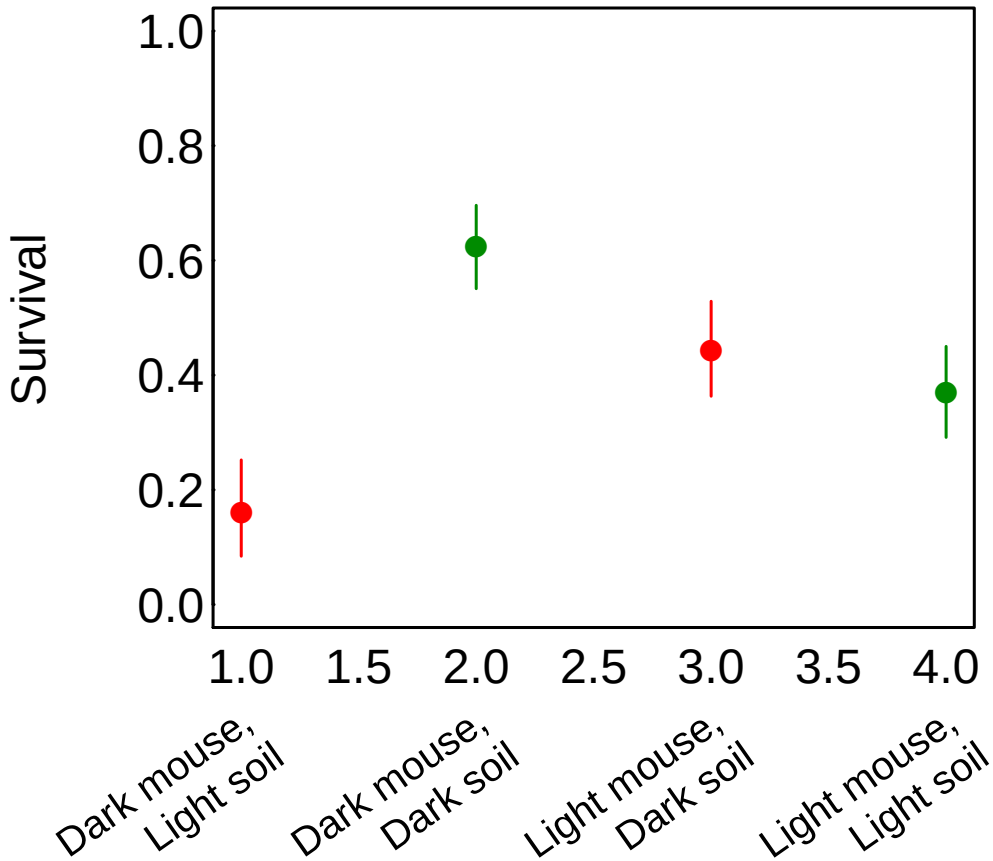
Whether a mouse is alive at time 'rep' depends on whether it was alive at time 'rep-1' and the probability of survival

Whether a mouse is seen at time 'rep' depends on whether it is alive and on the recapture rate


Posterior for phi.soil.phenotype
With 95%
quantile interval
Density
0.0
0.2
0.4
0.6
0.8
0
1
2
3
4
phi.soil.phenotype


Posterior for p.soil.phenotype
0.015
0.010
0.005
0.000
Density
−50
0
50
100
p.soil.phenotype

Survival of different mouse types in different enclosures



Quantities such as these can be calculated *a posteriori*, however, its easier to simply calculate them in your JAGS model and then track the calculated variable. That way you get chains, BCIs, etc.

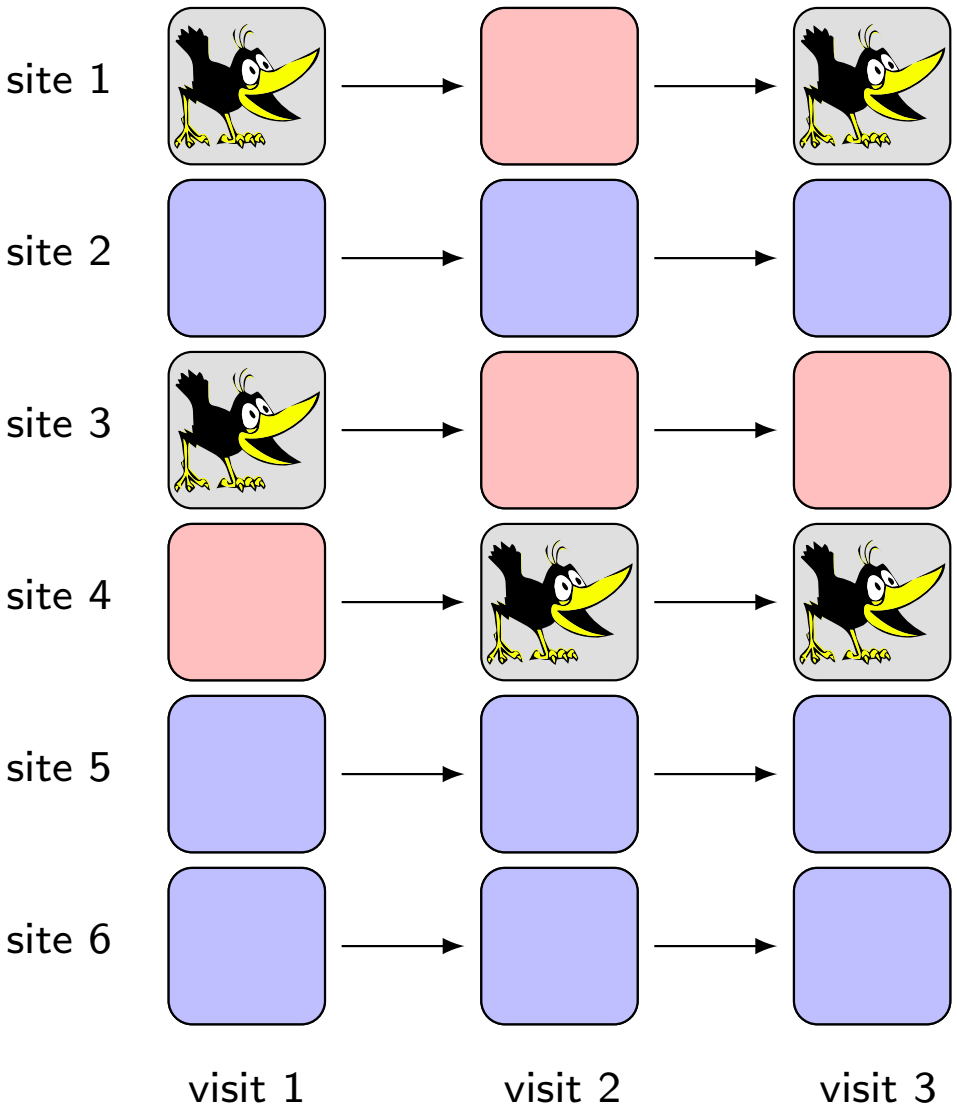
A species' *occupancy* is a measure of how many sites that species occupies.

Occupancy models extend the ideas from mark-recapture to make inferences about species' occupancy.

Instead of re-capturing individuals (as we did with mark-recapture), here we "re-capture" the species at a site if we detect that species at that site. For example, our survey method might be listening for bird calls. This is much less data-intensive than mark-recapture.

Mark-recapture models incorporate the imperfect "individual capture" process to improve inferences about total population size or rates of survival, recruitment, etc. Occupancy models incorporate the imperfect "species detection" process in order to improve inferences about the total number of occupied sites.

Suppose we survey 6 sites, 3 times each.

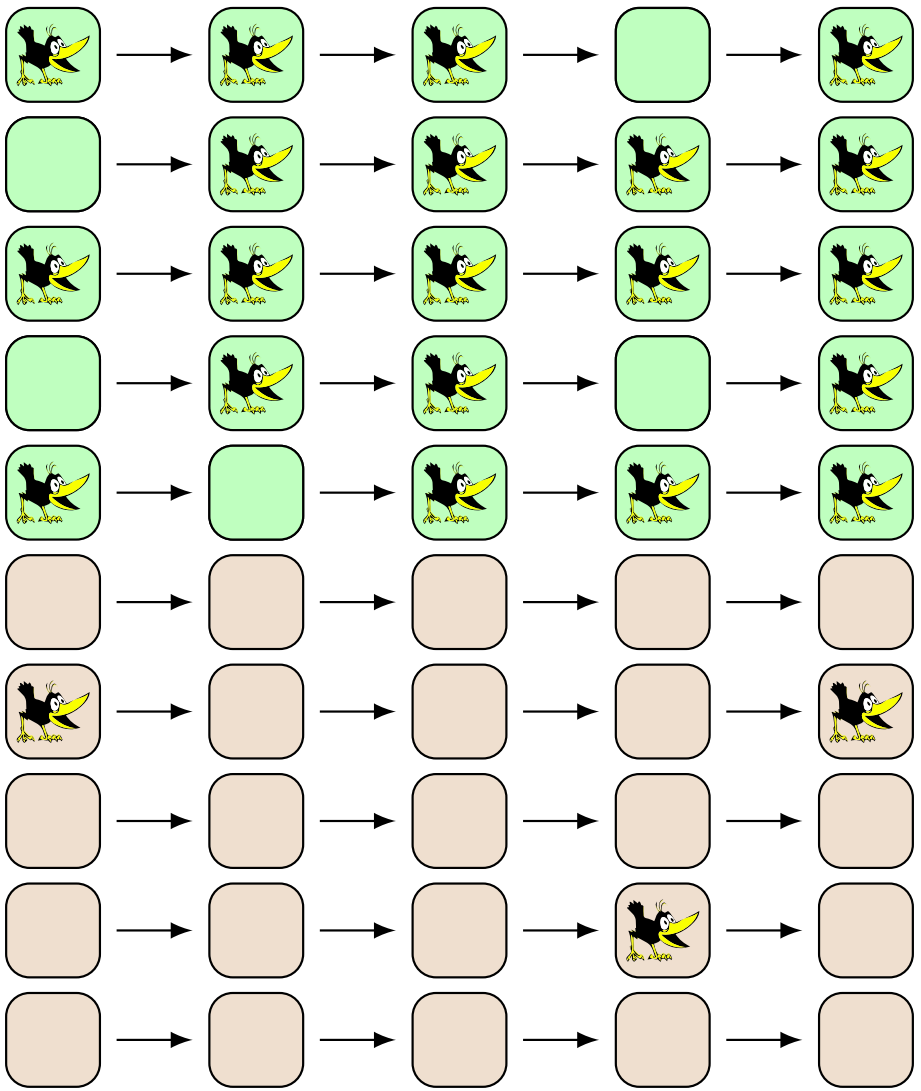


We know these are missed detections.

These could be missed detections, or the species might really be absent at these sites.

Just as we did for mark-recapture, we can include predictors on both rates of recapture and occupancy to look for effects of our predictors on these two processes.

Why do we need to do this?



Suppose we want to know whether green versus brown sites are better habitat for our favourite cartoon bird species? We survey birds over 5 occassions across 10 sites (5 green, 5 brown).

We find that 5 green sites and two brown sites contain the bird. Based on this, we conclude that brown sites are lower quality habitat for this bird. What is potentially wrong with this?

Accounting for imperfect recapture or species detection will greatly reduce your likelihood of making incorrect inferences.

Occupancy modeling is a powerful tool and one that is highly in demand!

Next week we will learn about multi-species occupancy models and dynamic multi-season occupancy models.

Survival Analysis: Involves estimating and comparing elapsed time until an event occurs.

https://www.emilyzabor.com/tutorials/survival_analysis_in_r_tutorial.html

GAMS:

http://environmentalcomputing.net/intro-to-gams

Replacement of fragstats in R:

https://r-spatialecology.github.io/landscapemetrics/

Good getting started vignette here

https://r-spatialecology.github.io/landscapemetrics/articles/getstarted.html