

# RHWO nest survival

*Anna Tucker*

*July 14, 2018*

Nest survival analysis for red-headed woodpeckers, dealing with uncertain nest fates. I'll walk through a few options, starting with the simplest survival model with censoring and up to a multievent model, with a Goldilocks option in between.

First load the packages we'll need. `tidyverse` contains my go-to functions for data wrangling and plotting, `cowplot` helps makes the plots look a little nicer, `lubridate` has functions for working with dates, and `survival` has functions for survival analysis. `jagsUI` will let us fit JAGS models from R.

```
knitr::opts_chunk$set(echo = T, include = T, message = F, warning = F)
require(tidyverse)
require(cowplot)
require(lubridate)
require(survival)
require(jagsUI)
```

## The data and questions

### Data collection

RHWO nests were monitored in the summer of 2016 from first discovery until they were determined to be successful (fledged at least one young). Due to logistical challenges it was not possible to observe the contents of some (most? all?) nests, so each visit was assigned a observation rank based on the activity observed around the nest.

#### Observation ranks:

- Rank 0 - No W detected
- Rank 1 - Single W in the area
- Rank 2 - Pair present
- Rank 3 - Excavating
- Rank 4 - Eggs
- Rank 5 - Chicks - Either bolus delivered, or chicks heard or seen in nest cavity
- Rank 6 - Juvi - Juvi detected in immediate area of nest cavity, usually parents of territory nearby

#### Nests were considered failed if:

- \* They didn't have fledged young about
- \* They were not active at least 36 days total
- \* They were not active with chicks for at least 24 days

### Objectives

Vegetation data was also collected for each nest, and the goal of this analysis is to determine the habitat characteristics that influence nest success for this species. More specially, to: 1. Estimate differences in nest failure (or success) between three general cover types 2. Evaluate the influence of individual covariates on nest failure, using some type of model selection process

### Challenges

The challenges of this particular data set:

- \* Nest age could not always be determined with certainty upon discovery
- \* Nest fate is uncertain for many nests
- \* Nest checks occurred at different intervals for each nest

## Exploratory data analysis

I saved the first sheet of the Excel file "Nest\_SummaryData.xlsx" as a csv. For now I'm only going to work with a subset of the columns:

WP\_ID unique nest ID Habitat.c general habitat type

D.Date discovery date

Last.Check.Date date when last nest check occurred

Monitoring.Duration total number of days the nest was observed

max.rank maximum observation rank in which it was observed

r0 rank on day of first discovery

Known.Success binary indicator of whether the nest was known to have fledged

Suspect Failed.c binary indicator whether nest was suspected to have failed (based on criteria above)

I rename them just to make the names a little shorter and easier to type, convert the dates to Julian days, fix a little habitat name typo, and create a nest fate column with value of either success, fail, or unknown.

```
dat = read_csv("data/rhwo-nest.csv")
head(dat)
```

```
## # A tibble: 6 x 70
##   WP_ID TA   Habitat.c Habitat `2ndary Habitat` `Edge <50m` Cav.Hgt
##   <chr> <chr> <chr>    <chr>    <chr>          <chr>    <chr>
## 1 NB 1A 01A   Wetland  Wetland <NA>         Closed Cano~ 11.2
## 2 PNB 1A 01A   Wetland  Wetland Snags         CC         11.6
## 3 NA 3A 03A   Wetland  Wetland Edge/young pine wo~ Pine woods 14.2
## 4 NB 3A 03A   Savannah Savannah Mixed         Woods      8.6
## 5 NB 3A2 03A   Wetland  Wetland Snags         Woods     10.4
## 6 NB 3A3 03A   Wetland  Wetland Snags         Woods      14
## # ... with 63 more variables: `Tree Hgt` <chr>, Taxon <chr>, `Bark` <chr>, Orient <chr>, `B.Crn?` <chr>, D.Date <chr>, in.0 <chr>,
## #   r.0 <int>, in.1 <chr>, r.1 <int>, in.2 <chr>, r.2 <int>, in.3 <chr>,
## #   r.3 <int>, in.4 <chr>, r.4 <int>, in.5 <chr>, r.5 <int>, in.6 <chr>,
## #   r.6 <int>, in.7 <chr>, r.7 <int>, in.8 <chr>, r.8 <int>, in.9 <chr>,
## #   r.9 <int>, `IN?` <chr>, M.Rank <int>, `First.R5+.Date` <chr>,
## #   Last.R5.Date <chr>, F.C.E <int>, F.C.E.adj <int>, D.C.E.adj <dbl>, `N
## #   duration with C.or.E` <int>, F.E <int>, `N duration with E` <int>,
## #   F.C <int>, F.C.adj <int>, D.C.adj <dbl>, `N duration with C` <int>,
## #   `First.R4+.Date` <chr>, Last.R4.Date <chr>, First.R6.Date <chr>,
## #   Last.Check.Date <chr>, D.Date_1 <chr>, Monitoring.Duration <int>,
## #   Last.Rank <int>, max.rank <int>, max.rank.c <chr>, Waypoint.ID <chr>,
## #   dccn <chr>, Not.a.nest <int>, Double.Brood <int>, Known.Success <int>,
## #   Prob.Success <int>, Prob.Success.c <int>, Prob.Success.Adj. <int>,
## #   Prob.Success.Adj.c <int>, Total_checks <int>, Avg.check.int <dbl>,
## #   `Suspect Failed` <dbl>, `Suspect Failed.c` <int>, `2 = classified as
## #   success and failure` <int>
```

```
dat_clean <- dat %>%
  select(WP_ID, Habitat.c, D.Date, Last.Check.Date, Monitoring.Duration,
         max.rank, r.0, Known.Success, `Suspect Failed.c`) %>%
  rename(id = WP_ID, hab = Habitat.c, init.date = D.Date, last.date = Last.Check.Date,
         obs.time = Monitoring.Duration, init.rank = r.0, max.rank = max.rank,
```

```

    success = Known.Success, fail = `Suspect Failed.c` ) %>%
mutate(init.date = parse_date(init.date, format = "%m/%d/%Y"),
      last.date = parse_date(last.date, format = "%m/%d/%Y"),
      init.day = yday(init.date),
      last.day = yday(last.date),
      hab = ifelse(hab == "woods", "Woods", hab),
      unknown.fate = ifelse(success == 0 & fail == 0, 1, 0),
      fate = case_when(
        success == 1 ~ "success",
        fail == 1 ~ "fail",
        unknown.fate == 1 ~ "unknown"
      )) %>%
select(-success, -fail, -unknown.fate)

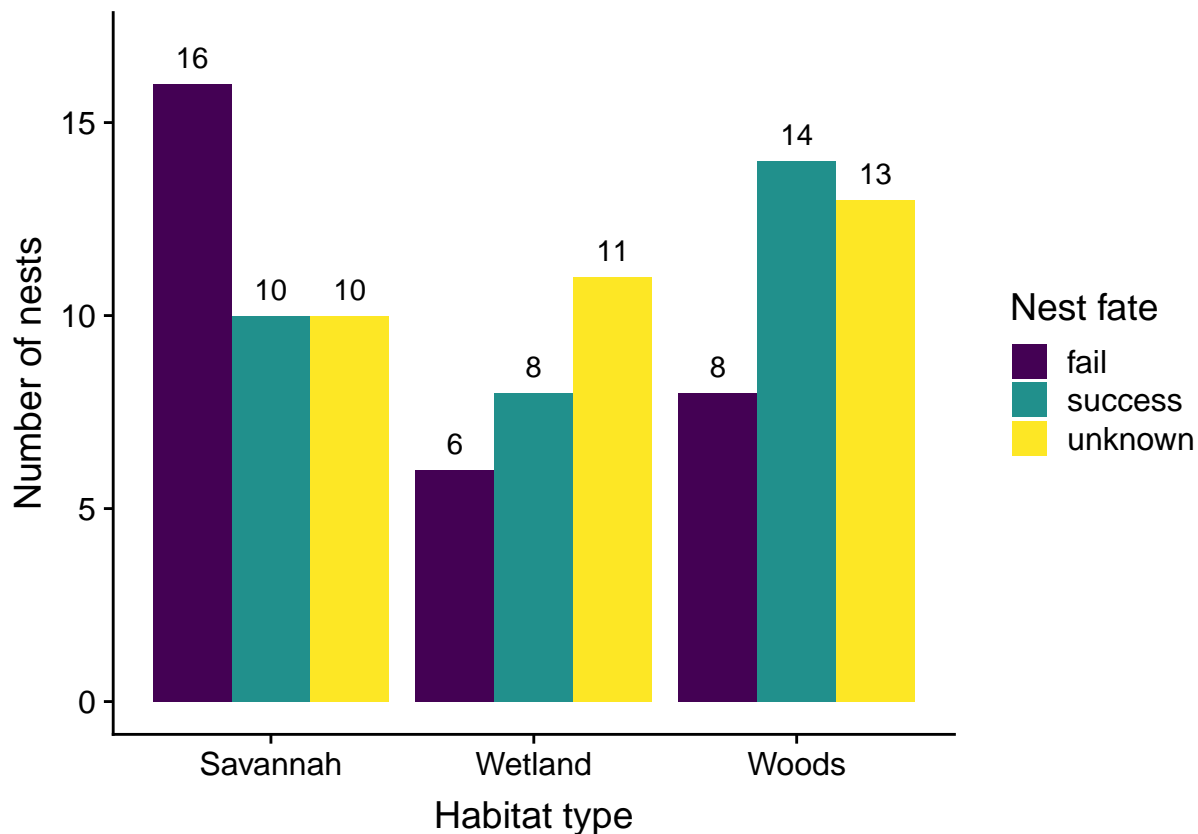
```

How many nests are in each fate in each habitat type?

```

dat_clean %>%
  group_by(hab) %>%
  count(fate) %>%
  ggplot(aes(x = hab, fill = fate, y = n)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label = n), position = position_dodge(width = 0.9),
    vjust = -0.75) +
  scale_fill_viridis_d(name = "Nest fate",
    labels = c("fail", "success", "unknown")) +
  xlab("Habitat type") +
  ylab("Number of nests") +
  ylim(c(0,17))

```



Looks like fairly equal sample sizes among habitat types.

```
dat_clean %>%
  group_by(hab) %>%
  count(fate) %>%
  spread(fate, n) %>%
  mutate(total = sum(fail, success, unknown))
```

```
## # A tibble: 3 x 5
## # Groups:   hab [3]
##   hab      fail success unknown total
##   <chr>   <int>   <int>   <int> <int>
## 1 Savannah    16     10     10    36
## 2 Wetland      6      8     11    25
## 3 Woods        8     14     13    35
```

```
dat_clean %>%
  count(fate)
```

```
## # A tibble: 3 x 2
##   fate      n
##   <chr>   <int>
## 1 fail     30
## 2 success  32
## 3 unknown  34
```

Roughly 1/3 of the nests have an unknown fate, but that's spread fairly evenly across habitat types, which is good.

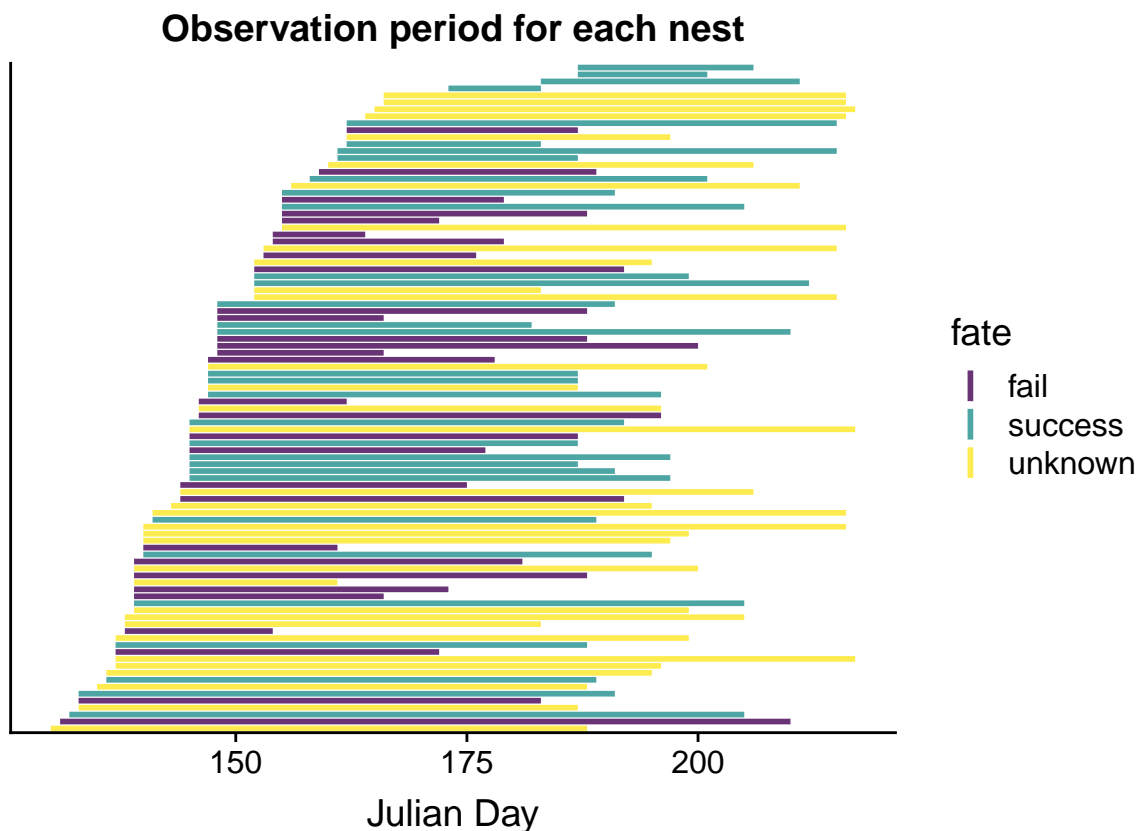
Quick check that the calculated monitoring duration matches up with the first and last date. (Asking R how many times do they not match up– 0 is the answer we’re looking for.)

```
length(which(dat_clean$last.day-dat_clean$init.day != dat_clean$obs.time))
```

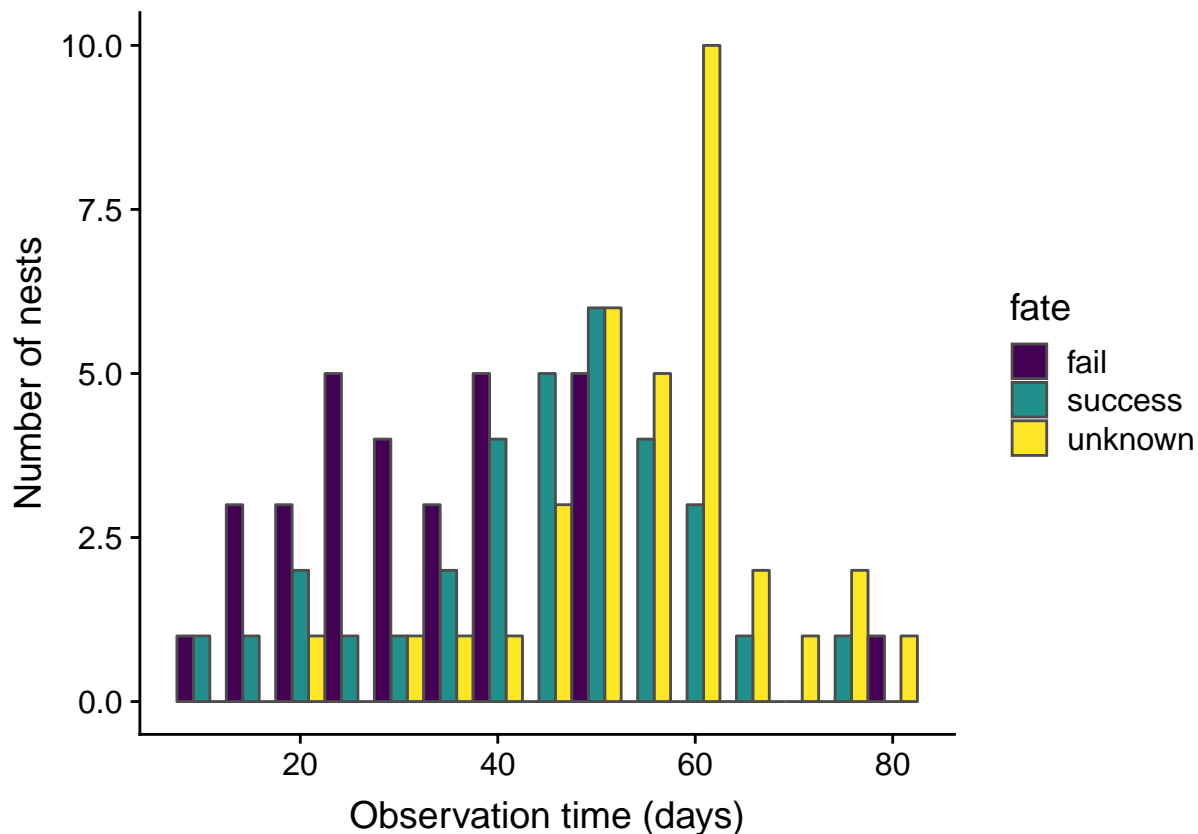
```
## [1] 0
```

Good. Let’s look at the observation times.

```
dat_clean %>%
  select(id, init.day, last.day, hab, fate) %>%
  mutate(id = as.factor(id),
         id = fct_reorder(id, init.day)) %>%
  ggplot(aes(x = id, col = fate)) +
  geom_linerange(aes(ymin = init.day, ymax = last.day), lwd = 1, alpha = 0.8) +
  scale_color_viridis_d() +
  ylab("Julian Day") +
  xlab("") +
  ggtitle("Observation period for each nest") +
  theme(axis.text.y = element_blank(),
        axis.ticks.y = element_blank()) +
  coord_flip()
```



```
ggplot(dat_clean, aes(x = obs.time, fill = fate)) +
  geom_histogram(col = "gray30", binwidth = 5, position = "dodge") +
  xlab("Observation time (days)") +
  ylab("Number of nests") +
  scale_fill_viridis_d()
```



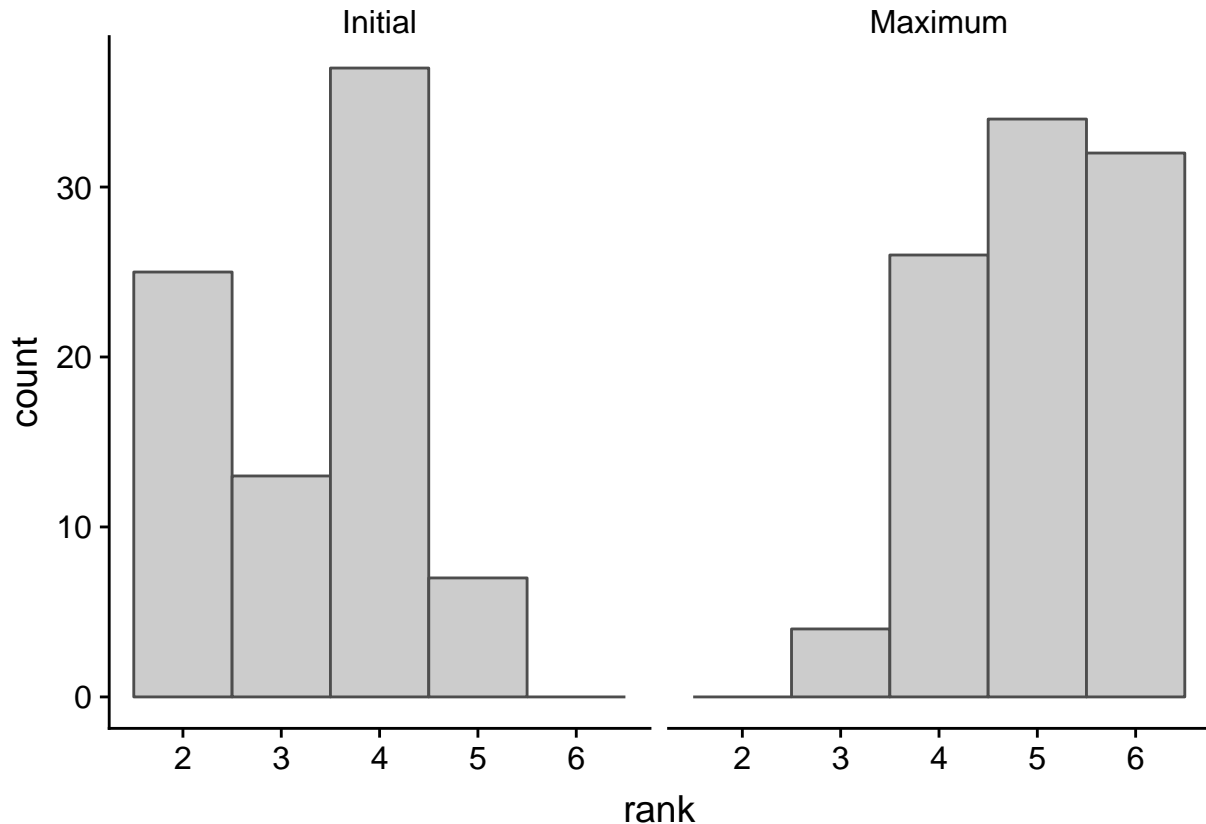
```
dat_clean %>%
  group_by(fate) %>%
  summarize(median.obs.time = median(obs.time))
```

```
## # A tibble: 3 x 2
##   fate   median.obs.time
##   <chr>         <dbl>
## 1 fail           31.5
## 2 success         47
## 3 unknown         56
```

Hmm so the successful nests have a Normal-ish distribution of observation times, while the failed nests were observed for shorter and the unknown fate nests observed for longer, which probably makes sense since you would only continue checking nests that you were unsure about. I don't think this is a "problem".

I'm also curious about the ranks.

```
dat_clean %>%
  select(init.rank, max.rank) %>%
  rename(Initial = init.rank, Maximum = max.rank) %>%
  gather(occ, rank, 1:2) %>%
  ggplot(aes(x = rank)) +
  geom_histogram(binwidth = 1, col = "gray30", fill = "gray80") +
  facet_wrap(~occ) +
  theme(strip.background = element_rect(fill = "white"))
```



What was the maximum rank of nests with each fate?

```
dat_clean %>%
  group_by(fate) %>%
  count(max.rank) %>%
  spread(fate, n)
```

```
## # A tibble: 4 x 4
##   max.rank fail success unknown
##   <int> <int>   <int>   <int>
## 1      3     3     NA       1
## 2      4    25     NA       1
## 3      5     2     NA     32
## 4      6    NA     32     NA
```

It looks like most of the failed nests failed in the egg stage, while most of the unknown fate nests made it to the chick stage.

### Analysis plan

Here I'll explore three different analysis options that increase in complexity. The first is classic survival-time-analysis/time-to-event analysis/proportional hazards. The second is a logistic exposure model, including a variation that uses information about the rank to help determine whether the nest failed. The last is a Bayesian multievent model.

## Proportional hazards model

The proportional hazards model estimates the probability of an event occurring (e.g. nest failure) as a function of time. Two common versions of this are Cox regression and Weibull regression. Using these models the nests with unknown fates would be censored (i.e. removed at some point).

### *Caveats/questions*

\* One challenge here is determining the day on which the censoring should occur.. we need to figure out the last day on which the nest was confirmed active, which could be a challenge. For this quick and dirty run through of the analysis I'm going to assume that the nest was known to be active up until the last day it was checked, but I suspect that's not correct for all nests. Interval censoring is also an option, but again would require us to define the days on which it was last confirmed active and first confirmed inactive.

\* We also need to account for the fact that nests were found at different initial stages. Is it possible to back-calculate nest initiation date with some degree of certainty?

Can we use information about the first day in ranks 4 and 5 to make some reasonable guesses at nest initiation date? I based this on your notes that it is 36 days on average from initiation to fledge, and 12 days average incubation period. This is a **very rough** approximation, just to play around with how to use this information in the model. I'm calculating nest initiation day based on the very strong and unreasonable assumptions that the first day chicks are observed is hatch day, and saying that if the maximum rank ever observed was eggs, then the nest was found on the day it was initiated.

```
dat_surv <- dat %>%
  select(WP_ID, D.Date, Last.Check.Date, `First.R4+.Date`, `First.R5+.Date`, r.0, max.rank) %>%
  rename(id = WP_ID, found = D.Date, last = Last.Check.Date, init.rank = r.0,
         first.eggs = `First.R4+.Date`, first.chicks = `First.R5+.Date`) %>%
  gather(event, date, 2:5) %>%
  mutate(date = parse_date(date, format = "%m/%d/%Y")) %>%
  spread(event, date) %>%
  mutate(est.NI = case_when(
    max.rank > 4 ~ first.chicks-12,
    max.rank == 4 ~ found)) %>%
  full_join(dat_clean) %>%
  mutate(failed = ifelse(fate == "fail", 1, 0),
         time.to.fail = as.numeric(last - est.NI))
```

## Cox regression

Here I'm going to frame this as time-to-failure, but we could easily run it as time-to-success instead.

For this analysis the data needs to be organized with the following information in columns:

\* number of days until the event (here, nest failure or last known active date) occurred

\* binary indicator of whether the nest failed, this equals 1 if failure occurred and 0 if the fate was unknown.

All other nests are censored

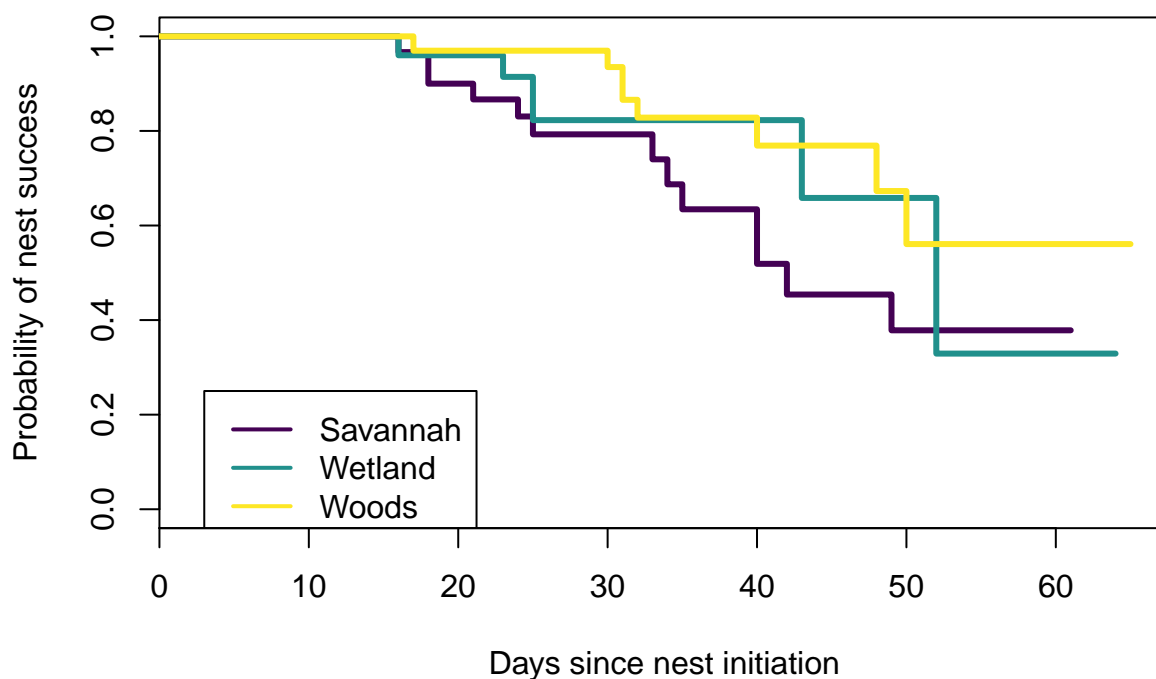
The functions in the `survival` function use the survival object.

```
dat_surv$survobj = with(dat_surv, Surv(time.to.fail, failed))
```

A quick look at the Kaplan-Meier curve for all nests, assuming equal status upon first discovery for now.

```
plot(survfit(survobj ~ hab, data = dat_surv), xlab = "Days since nest initiation",
     ylab = "Probability of nest success", col = viridis::viridis(3), lwd = 3)
legend(3, 0.25, lwd = rep(2, 3), col = viridis::viridis(3), c("Savannah", "Wetland", "Woods"))
```





Summary of Cox regression, where Savannah is the reference level.

```
summary(coxmod <- coxph(survobj ~ hab, data = dat_surv))
```

```
## Call:
## coxph(formula = survobj ~ hab, data = dat_surv)
##
##   n= 92, number of events= 27
##   (4 observations deleted due to missingness)
##
##              coef exp(coef) se(coef)      z Pr(>|z|)
## habWetland -0.4103   0.6634  0.4958 -0.828  0.4079
## habWoods   -0.7646   0.4655  0.4500 -1.699  0.0893 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##              exp(coef) exp(-coef) lower .95 upper .95
## habWetland    0.6634      1.507    0.2511    1.753
## habWoods      0.4655      2.148    0.1927    1.125
##
## Concordance= 0.611 (se = 0.06 )
## Rsquare= 0.032 (max possible= 0.892 )
## Likelihood ratio test= 3.01 on 2 df,  p=0.2
## Wald test               = 2.95 on 2 df,  p=0.2
## Score (logrank) test = 3.07 on 2 df,  p=0.2
```

```
AIC(coxmod)
```

```
## [1] 206.1625
```

This indicates that both Wetlands and Woods have a lower hazard rate (i.e. nest failure rate) than Savannah ( $\exp(\text{coef}) < 1$ ). A nest in wetlands has a reduction in risk of nest failure of 0.66, or 34% lower. A nest in the woods cover type has a 0.46 lower risk of nest failure compared to savannah, or 54% lower risk of nest failure.

## Weibull regression

Compare that to estimates from the Weibull regression, which also estimates a scale parameter  $p$  that describes how the hazard rate changes over time. When  $p > 1$ , hazard (risk of failure) increases over time, and when  $p < 1$  hazard decreases over time.

```
summary(weibmod <- survreg(survobj ~ hab, data = dat_surv, dist = "weibull"))
```

```
##
## Call:
## survreg(formula = survobj ~ hab, data = dat_surv, dist = "weibull")
##              Value Std. Error      z      p
## (Intercept)  3.951      0.107 36.95 < 2e-16
## habWetland   0.154      0.186  0.83  0.409
## habWoods     0.301      0.171  1.76  0.079
## Log(scale)  -0.990      0.153 -6.49 8.6e-11
##
## Scale= 0.372
##
## Weibull distribution
## Loglik(model)= -141.2  Loglik(intercept only)= -142.9
##  Chisq= 3.37 on 2 degrees of freedom, p= 0.19
## Number of Newton-Raphson Iterations: 7
## n=92 (4 observations deleted due to missingness)
```

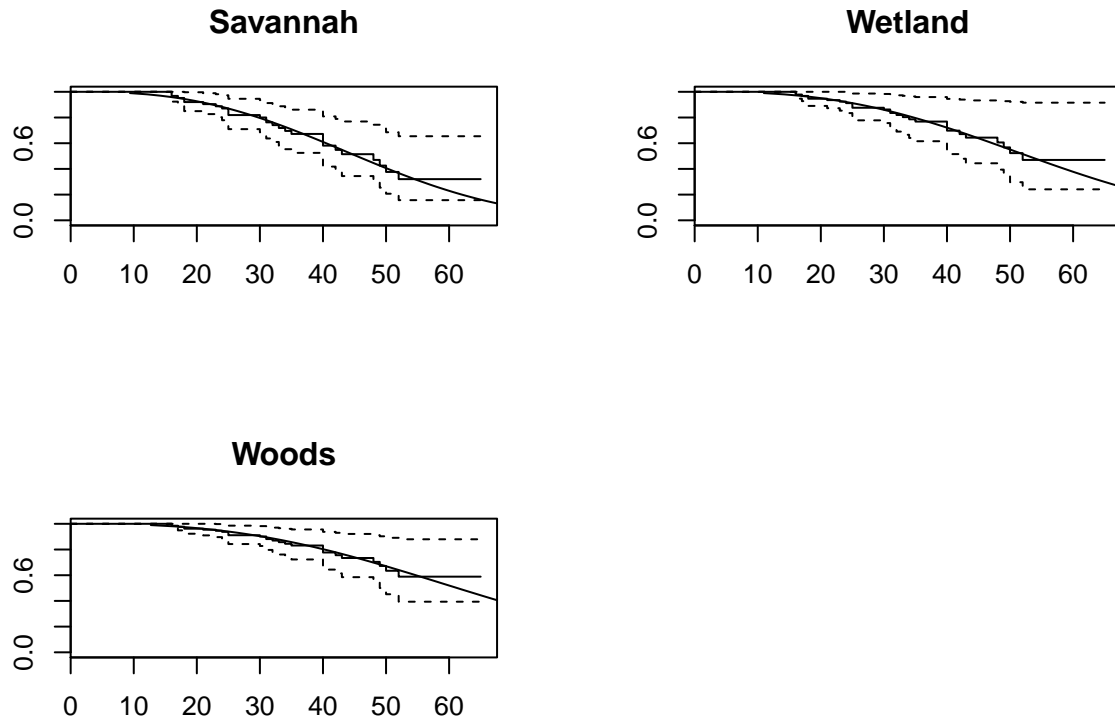
```
AIC(weibmod)
```

```
## [1] 290.4987
```

Compare predictions from the two models. The Cox model is the jagged line and the Weibull regression is smooth.

```
habitat = c("Savannah", "Wetland", "Woods")

par(mfrow = c(2, 2))
for(i in 1:3){
  plot(survfit(coxmod, newdata = data.frame(hab = habitat[i])), main = habitat[i])
  lines(x = predict(weibmod, newdata = data.frame(hab = habitat[i]), type = "quantile",
    p = seq(0.01, 0.99, by=.01)), y = rev(seq(0.01, 0.99, by = 0.01)))
}
```



I'm not sure how we would decide between the Cox and Weibull regressions, to be honest. They produce very similar estimates.

### Advantages and disadvantages

#### *Pros*

- \* Easy to censor uncertain fates
- \* AIC model selection straightforward to implement
- \* Established in the literature

#### *Cons*

- \* Both unknown-fate nests and those known to survive are censored—this feels like throwing information away about known successes?
- + I recommend we analyze both time-to-failure and time-to-success and present both to get a fuller picture
- \* Need to determine nest initiation date in some kind of reasonable/defensible way
- \* Need to determine the date of the “event” and date of censoring
- + interval censoring might be a good option, where we define the date at which we last knew it was active and the first date at which we knew it had failed/fledged or stopped monitoring

### Logistic exposure model

The basic idea here is that we use a logistic regression to model success/failure (1/0) as a function of covariates. The difference with regular logistic regression is the link function, which instead of being a normal logit:

$$\log_e \left( \frac{p}{1-p} \right) = \beta_0 + \beta_1 * x_1 + \dots$$

includes a term for exposure time, i.e. the number of days the nest has been active:

$$\log_e \left( \frac{p^{1/t}}{1 - p^{1/t}} \right) = \beta_0 + \beta_1 * x_1 + \dots$$

Where  $p$  is the probability of surviving the interval, and  $p = s^t$  where  $s$  is the daily survival probability. Thanks to Ben Bolker, this can be easily implemented with the standard GLM function in R by using this custom link function code from here:

```
logexp <- function(exposure = 1){
  linkfun <- function(mu) qlogis(mu^(1/exposure))
  linkinv <- function(eta) plogis(eta)^exposure
  logit_mu_eta <- function(eta) {
    ifelse(abs(eta)>30,.Machine$double.eps,
           exp(eta)/(1+exp(eta))^2)
  }
  mu.eta <- function(eta) {
    exposure * plogis(eta)^(exposure-1) *
    logit_mu_eta(eta)
  }
  valideta <- function(eta) TRUE
  link <- paste("logexp(", deparse(substitute(exposure)), ")",
               sep="")
  structure(list(linkfun = linkfun, linkinv = linkinv,
                 mu.eta = mu.eta, valideta = valideta,
                 name = link),
            class = "link-glm")
}
```

And instead of using

```
glm(...family = binomial(link = logit))
```

you just use

```
glm(...family = binomial(link = logexp(exposure.time)))
```

and provide exposure time as data for each nest.

For a typical logistic exposure model, you need to have known success or failure for every nest, so that is not really an option here. However, I think we can get around that by using a hierarchical model. Essentially we take the basic logistic exposure model:

$$\text{logit.exp}(p_i) = \beta_0 + \beta_1 * x_1 + \dots \text{success}_i \sim \text{Binomial}(1, p_i)$$

And add another level for the observation process. We define a matrix that describes the probability of observing nests in a maximum rank (3, 4, 5, or 6) given its true fate (failed, successful). The rows indicate the true state and the columns are the observations.

$$\text{obs} = \begin{bmatrix} r_{F,3} & r_{F,4} & r_{F,5} & r_{F,6} \\ r_{S,3} & r_{S,4} & r_{S,5} & r_{S,6} \end{bmatrix}$$

For example,  $r_{F,3}$  is the probability of a failed nest having a maximum observed rank of 3. All of the rows of this matrix must sum to 1, because all nests must have some observed maximum rank.

We estimate those probabilities using the Multinomial distribution, which is a cousin of the Binomial for when there are multiple possible discrete outcomes. The observed maximum rank is denoted with  $y$  and  $s_i$  is an indicator of success or failure, where  $s_i = 1$  for failed nests and  $s_i = 2$  for successful nests.

$$y_i \sim \text{Multinomial}(\text{obs}[s_i,])$$

Instead of using success/failure as our dependent variable, the observed maximum rank becomes the ultimate dependent variable, and success/failure is treated like something in between data and a parameter. We provide 0s and 1s for nests with known successes and failures and NAs for nests with unknown fates. The model uses the known fate cases to estimate the observation probabilities and regression coefficients, which are then used along with the information in regression covariates to probabilistically assign nests as successes and failures.

## JAGS model

I don't know how to fit this model using MLE, but we can fit it using JAGS.

```
cat(file = "log_exp.txt", "
model{

for(i in 1:n){
  z[i] <- alpha + beta[hab[i]]
  p[i] <- pow((1/(1 + exp(-z[i]))), t[i])
  s[i] ~ dbern(p[i])
  srow[i] <- s[i] + 1
  y[i] ~ dcat(obs[srow[i],])
}

for(i in 1:2){
  obs[i,1:4] ~ ddirch(dpriors)
}

alpha ~ dnorm(0, 0.1)

for(i in 1:3){
  beta[i] ~ dnorm(0, 0.1)
}

}
")
```

I always test a new model with simulated data, to make sure it's doing what I think it's doing. Here I simulate data for 100 nests in one of three habitat types, that have been observed for up to 30 days. I then choose 30 nests randomly and assign their fate as NA.

```
alpha = 1
beta = c(-2, 0, 2)

n = 100
x = rnorm(n, 0, 5)
hab = sample(c(1:3), n, replace = T)
t = runif(n, 2, 10)

obs = matrix(c(
  0.25, 0, 0, 0.75,
```

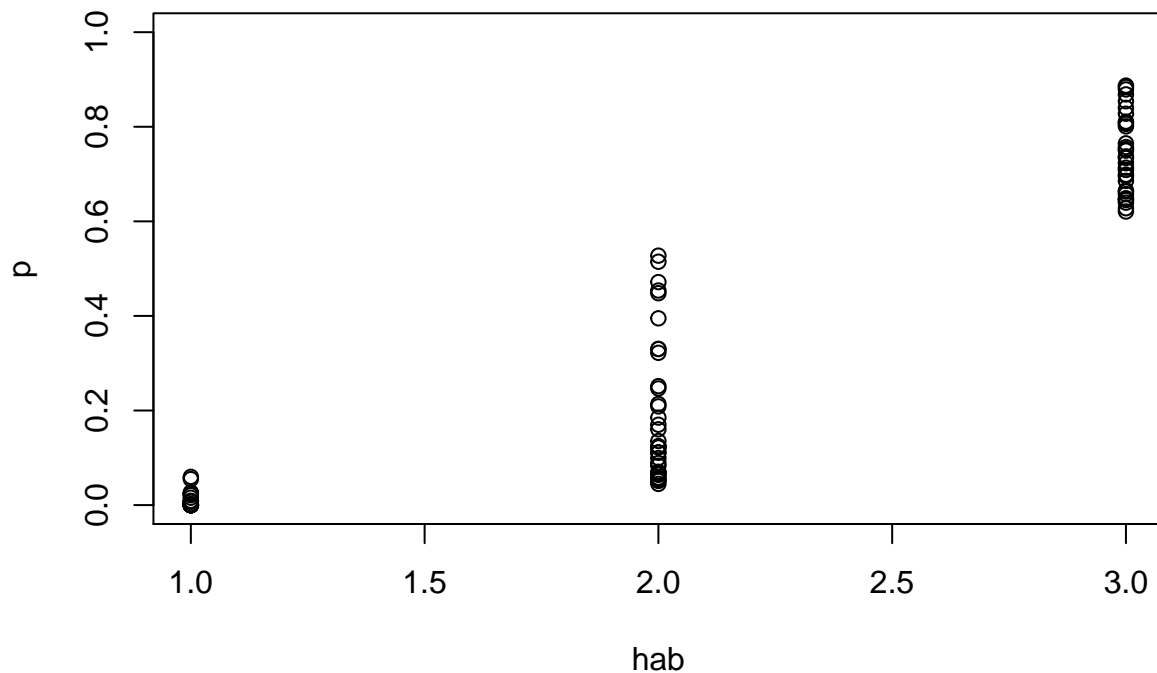
```

0.5, 0.2, 0.2, 0.1), nrow = 2, ncol = 4, byrow = T)

linpred = p = s = y = srow = numeric()
for(i in 1:n){
  linpred[i] = alpha + beta[hab[i]]
  p[i] = plogis(linpred[i])^t[i]
  s[i] = rbinom(1, 1, p[i])
  srow[i] = s[i]+1
  y[i] = which(rmultinom(1, 1, obs[srow[i],]) == 1)
}
to.remove = sample(1:n, 30, replace = T)
obs.s = s
obs.s[to.remove] = NA

plot(p ~ hab, ylim = c(0, 1))

```



```
table(s, y)
```

```

##      y
## s    1  2  3  4
## 0 19  0  0 48
## 1 16  3  8  6

```

Next we bundle the data for JAGS, tell it what parameters we want output for, and define the MCMC estimation parameters.

```
jagsdat = list(s = obs.s,
              t = t,
              y = y,
              hab = hab,
              dpriors = rep(1,4),
              n = length(s))

parms = c("alpha", "beta", "obs")

nc = 3
ni = 50000
nb = 10000
na = 10000
nt = 50

test = jags(jagsdat, inits = NULL, parameters.to.save = parms,
            model.file = "log_exp.txt",
            n.chains = nc, n.iter = ni, n.burnin = nb, n.thin = nt,
            n.adapt = na, parallel = T)
```

```
##
## Processing function input.....
##
## Done.
##
## Beginning parallel processing using 3 cores. Console output will be suppressed.
##
## Parallel processing completed.
##
## Calculating statistics.....
##
## Done.
```

Model output:

```
test$summary
```

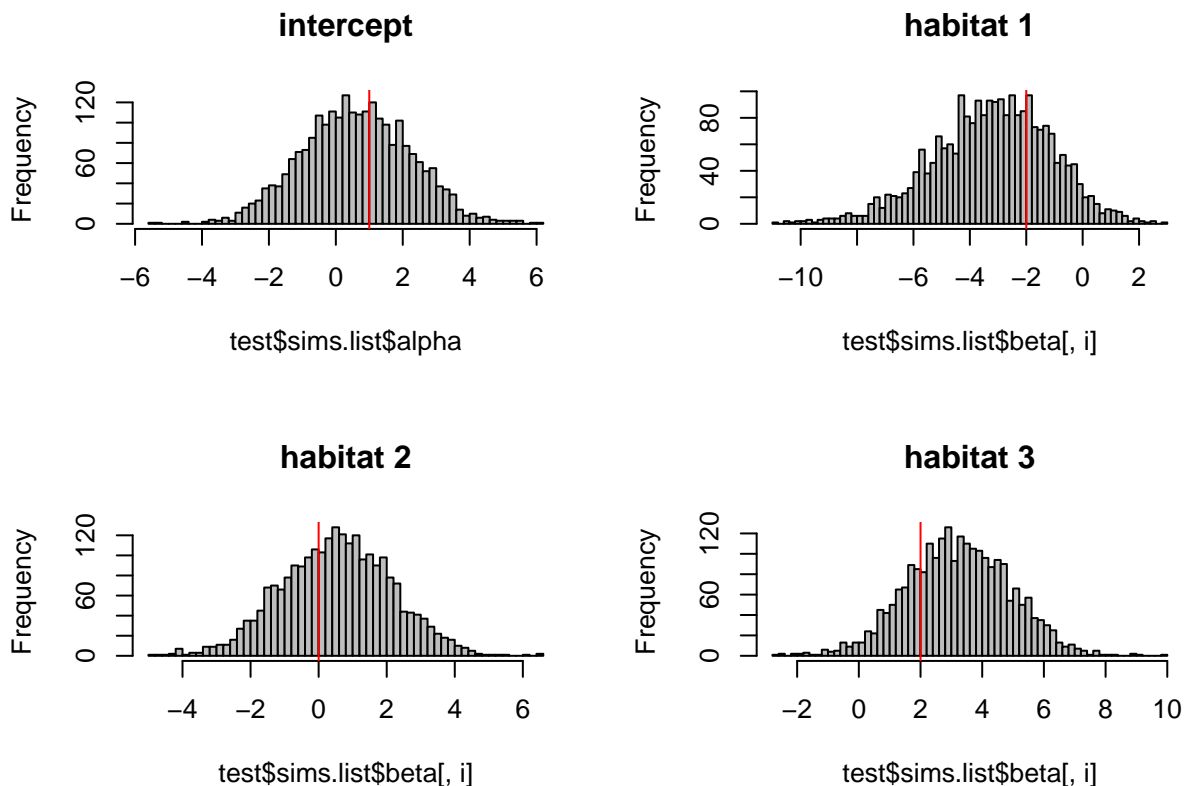
| ##          | mean         | sd         | 2.5%          | 25%           | 50%          |
|-------------|--------------|------------|---------------|---------------|--------------|
| ## alpha    | 0.61294127   | 1.61443105 | -2.455876e+00 | -0.484887601  | 0.59769066   |
| ## beta[1]  | -3.21405140  | 2.09140063 | -7.582540e+00 | -4.499464112  | -3.10464713  |
| ## beta[2]  | 0.53509400   | 1.61434194 | -2.589999e+00 | -0.576048712  | 0.56040820   |
| ## beta[3]  | 3.22942548   | 1.67401973 | -1.532447e-02 | 2.081991405   | 3.21226041   |
| ## obs[1,1] | 0.28932922   | 0.05717666 | 1.860907e-01  | 0.248136906   | 0.28708324   |
| ## obs[2,1] | 0.43982994   | 0.08215008 | 2.837076e-01  | 0.383017328   | 0.43789660   |
| ## obs[1,2] | 0.01552624   | 0.01513536 | 3.615589e-04  | 0.004768556   | 0.01087161   |
| ## obs[2,2] | 0.10108839   | 0.04865511 | 2.747322e-02  | 0.065511671   | 0.09314071   |
| ## obs[1,3] | 0.01483351   | 0.01483984 | 4.105727e-04  | 0.004184349   | 0.01000463   |
| ## obs[2,3] | 0.22497758   | 0.06571233 | 1.107299e-01  | 0.176304101   | 0.22135275   |
| ## obs[1,4] | 0.68031103   | 0.05813702 | 5.588709e-01  | 0.641848825   | 0.68312554   |
| ## obs[2,4] | 0.23410408   | 0.07228575 | 1.093763e-01  | 0.182901600   | 0.22983728   |
| ## deviance | 222.20456416 | 5.63123471 | 2.123172e+02  | 218.270955901 | 221.84193217 |
| ##          | 75%          | 97.5%      | Rhat          | n.eff         | overlap0     |
| ## alpha    | 1.75164544   | 3.65540842 | 1.0042121     | 439           | 1 0.6416667  |
| ## beta[1]  | -1.76077913  | 0.50992758 | 1.0001296     | 2400          | 1 0.9533333  |
| ## beta[2]  | 1.66223090   | 3.63744981 | 1.0043044     | 435           | 1 0.6325000  |

```
## beta[3]      4.39268149    6.37936132 1.0061566    305        1 0.9741667
## obs[1,1]    0.32831298    0.41210094 1.0033167    518        0 1.0000000
## obs[2,1]    0.49659707    0.60067546 0.9994916   2400        0 1.0000000
## obs[1,2]    0.02154527    0.05823225 1.0084743    567        0 1.0000000
## obs[2,2]    0.13108113    0.21637409 1.0003165   2400        0 1.0000000
## obs[1,3]    0.02117796    0.05223595 1.0013714   2400        0 1.0000000
## obs[2,3]    0.26701662    0.36392418 1.0001443   2400        0 1.0000000
## obs[1,4]    0.72164241    0.78806914 1.0018715    827        0 1.0000000
## obs[2,4]    0.28053730    0.39034931 0.9999188   2400        0 1.0000000
## deviance 225.69524903 234.15734391 1.0012241   1282        0 1.0000000
```

Posterior distributions for intercept and habitat effects (the red line is the true value that we used to simulate the data):

```
par(mfrow = c(2,2))
hist(test$sims.list$alpha, breaks = 50, col = "gray", main = "intercept")
abline(v = alpha, col = "red")

for(i in 1:3){
  hist(test$sims.list$beta[,i], breaks = 50, col = "gray", main = paste("habitat", i))
  abline(v = beta[i], col = "red")
}
```



Posterior distributions for the probabilities of observing each maximum rank given true success or failure:

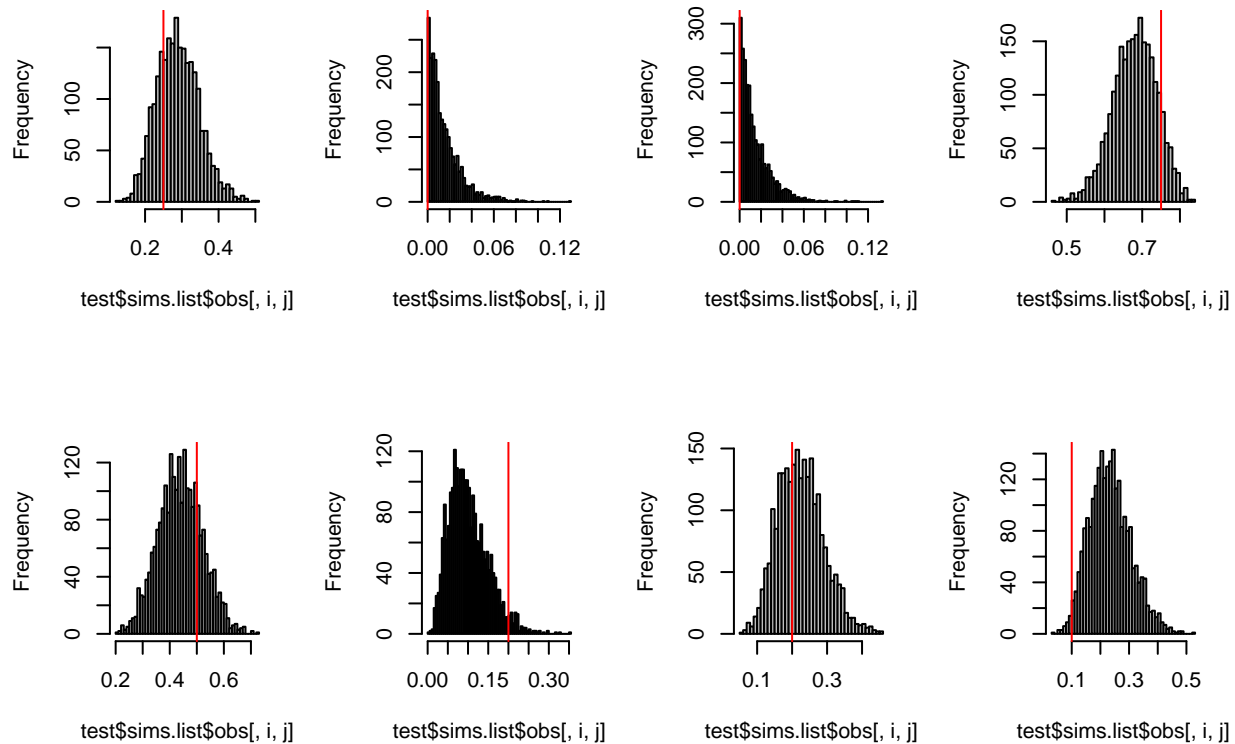
```
par(mfrow = c(2,4))
for(i in 1:2){
  for(j in 1:4){
```



```

hist(test$sims.list$obs[,i,j], col = "gray", breaks = 50, main = "")
abline(v = obs[i,j], col = "red")
}
}

```



It looks like the model works pretty well at recovering the regression coefficients even with missing data. However, the estimates are not very precise. I suspect precision would increase if we had more data, but I designed the simulation to be similar to the data we have.

Let's take a quick look at fitting this model with the RHW0 data. Maximum rank is from 3-6, so I rescale it from 1-4 for the analysis. I also recode habitat as numeric, where 1 = savannah, 2 = wetlands, and 3 = woods. There are a few NAs for exposure time.. since the way I calculated that will probably be redone anyway I'm not going to bother looking into those now and just assign any NAs to equal the median exposure time.

```

dat_surv %>%
  mutate(s = case_when(
    fate == "fail" ~ 0,
    fate == "success" ~ 1,
  )) %>%
  pull(s) -> s

dat_surv %>%
  pull(time.to.fail) -> t
t[which(is.na(t))] = median(t, na.rm = T)

dat_surv %>%
  mutate(obs = max.rank - 2) %>%

```

```

pull(obs) -> y

dat_surv %>%
  mutate(habnum = case_when(
    hab == "Savannah" ~ 1,
    hab == "Wetland" ~ 2,
    hab == "Woods" ~ 3)) %>%
  pull(habnum) -> hab

jagsdat = list(s = s,
               t = t,
               y = y,
               hab = hab,
               dpriors = rep(1,4),
               n = length(s))

parms = c("alpha", "beta", "obs")

nc = 3
ni = 50000
nb = 10000
na = 10000
nt = 50

rhwo1 = jags(jagsdat, inits = NULL, parameters.to.save = parms,
             model.file = "log_exp.txt",
             n.chains = nc, n.iter = ni, n.burnin = nb, n.thin = nt,
             n.adapt = na, parallel = T)

##
## Processing function input.....
##
## Done.
##
## Beginning parallel processing using 3 cores. Console output will be suppressed.
##
## Parallel processing completed.
##
## Calculating statistics.....
##
## Done.

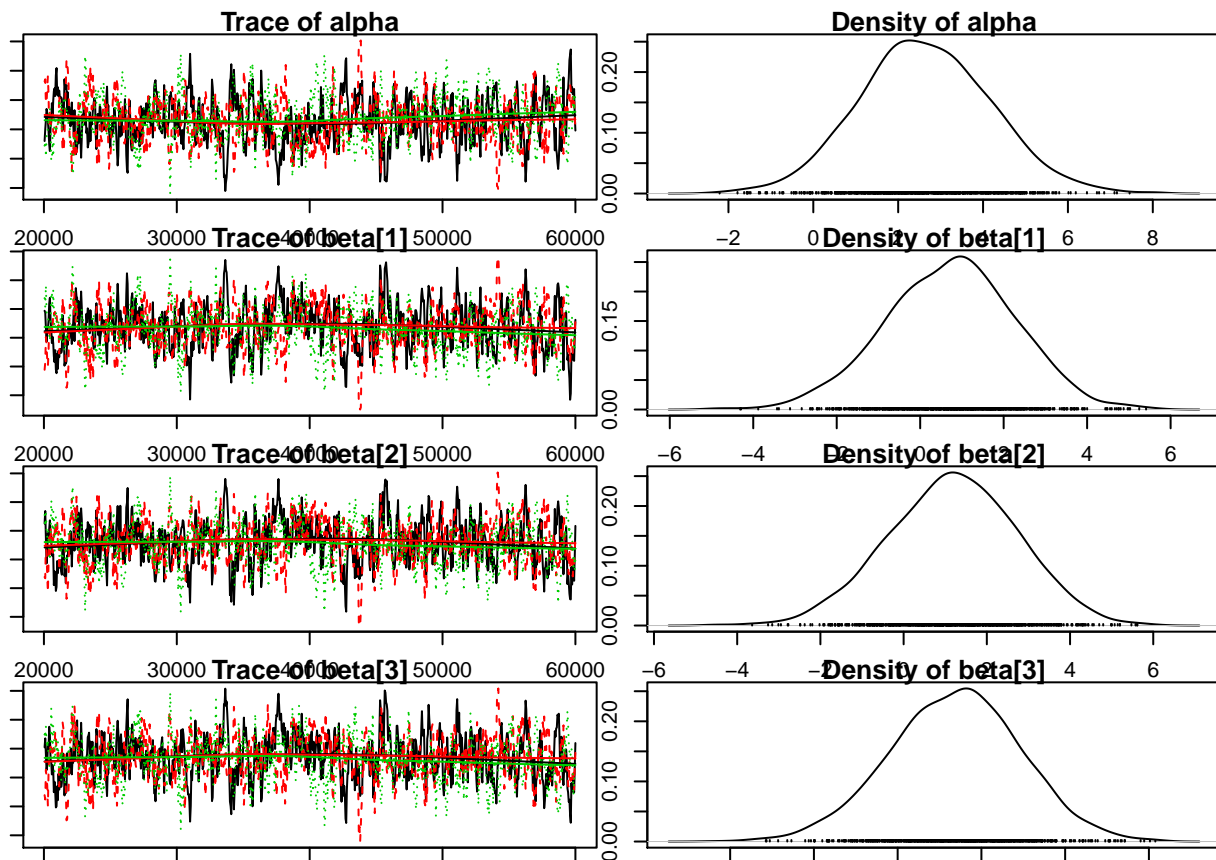
```

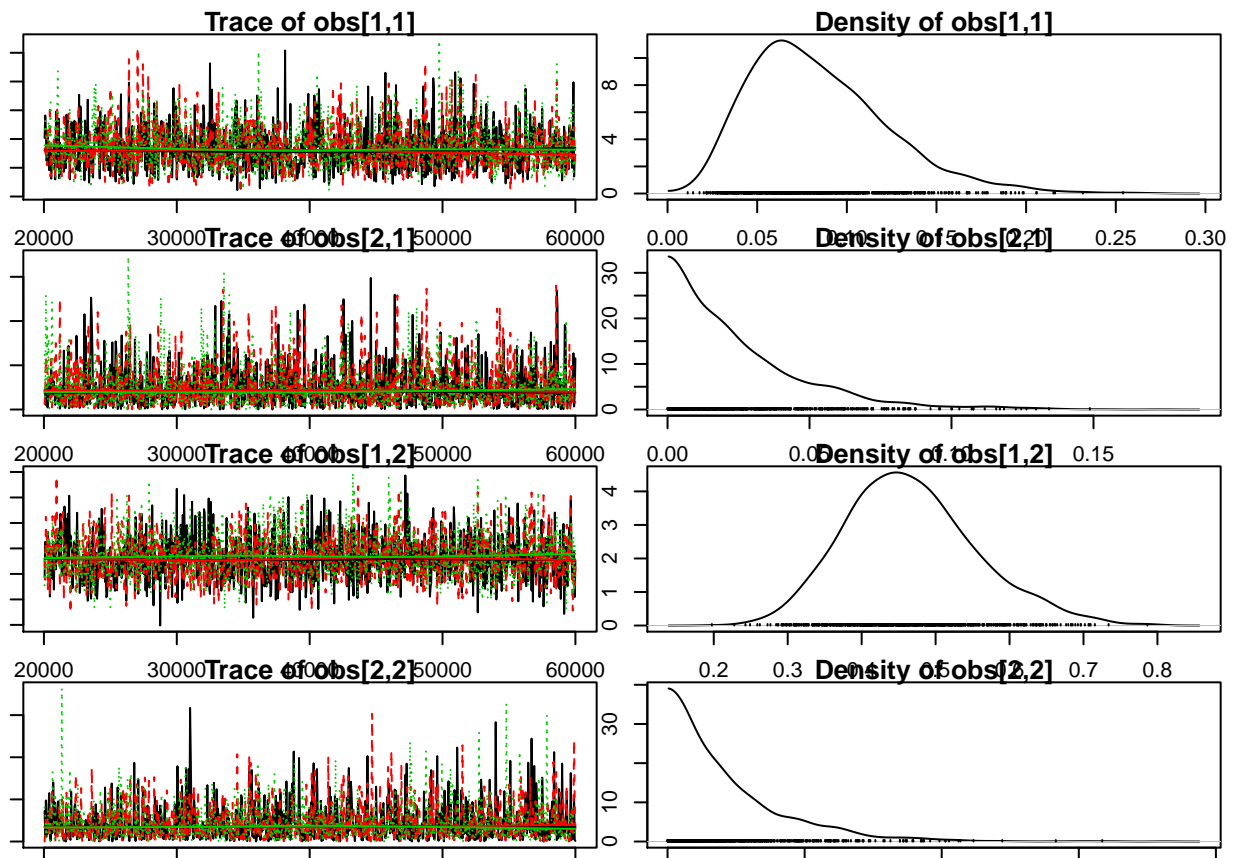
Check convergence (we're looking for nice normal-ish looking distributions and “fuzzy” traceplots where all the colors overlap):

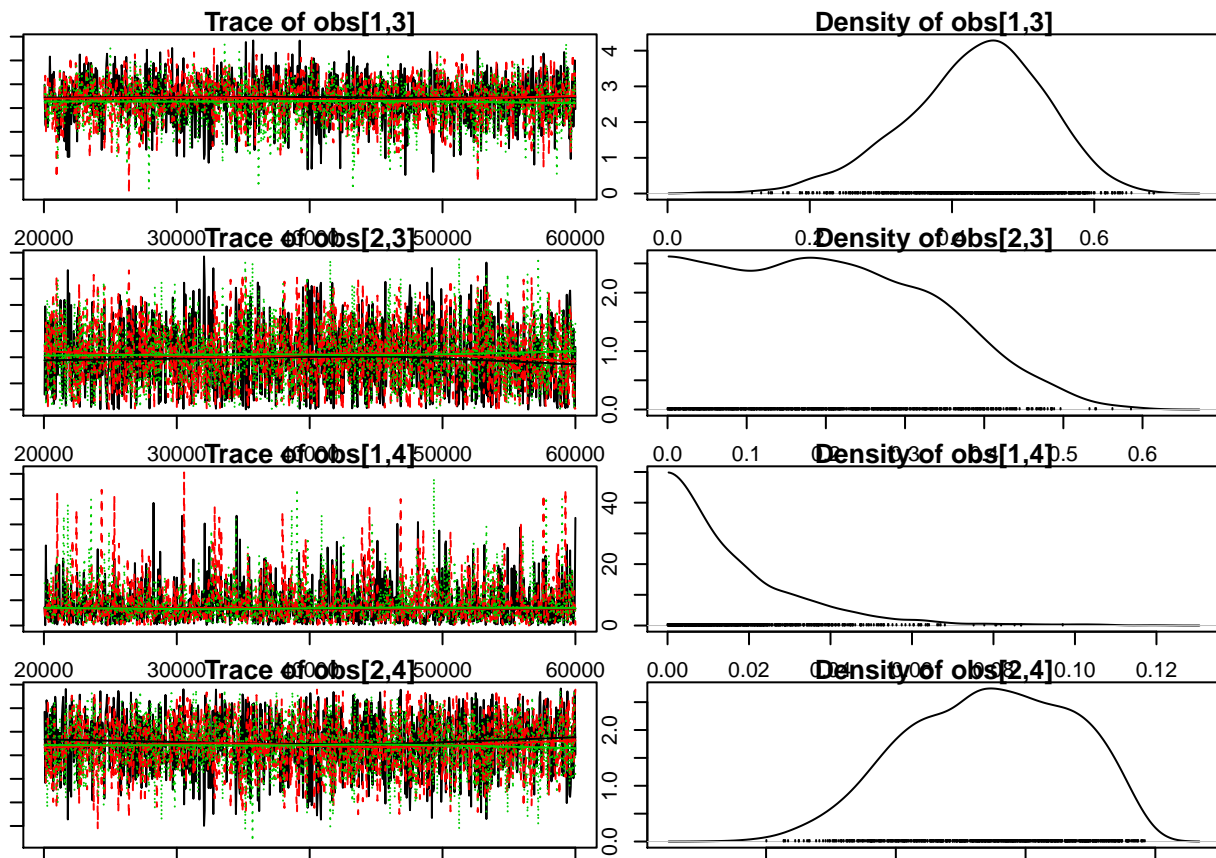
```

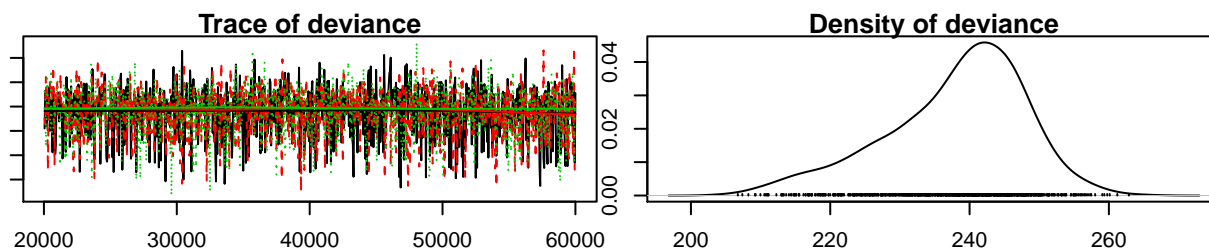
par(mar = c(1,1,1,1))
plot(rhwo1)

```









Look at model outputs:

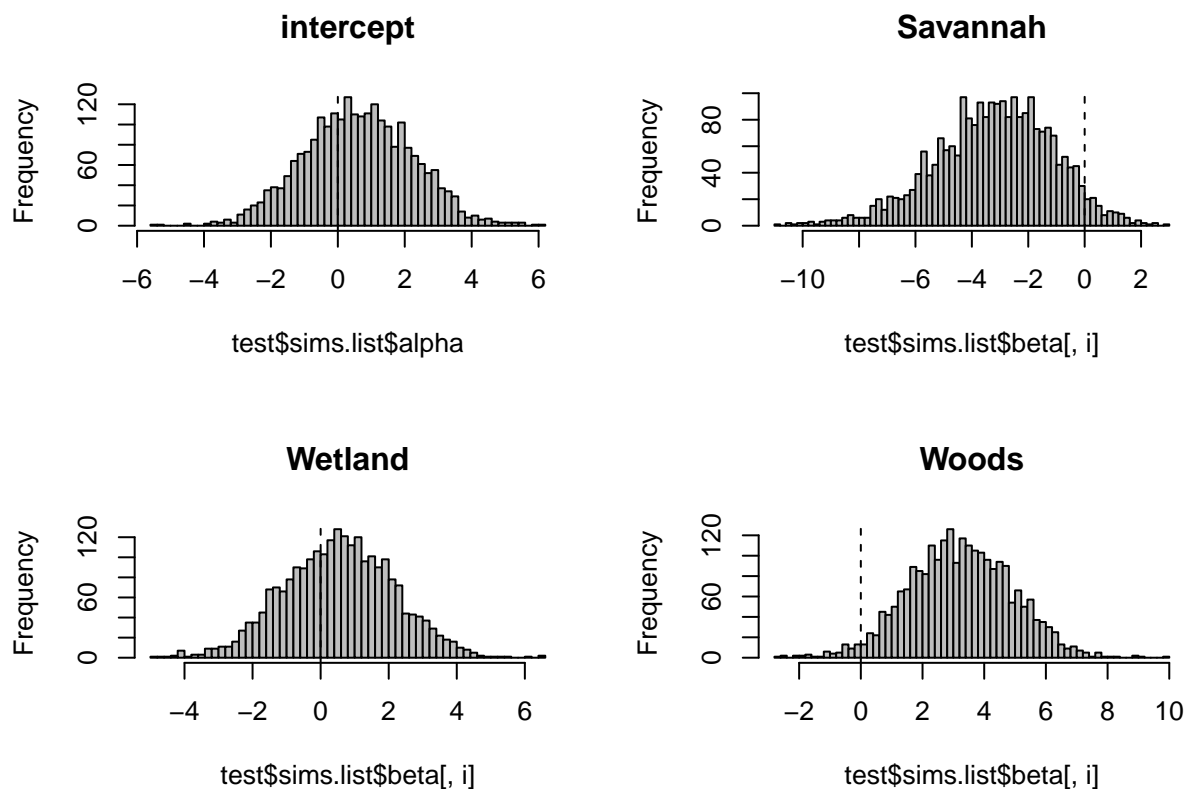
```
rhwo1$summary
```

| ## |          | mean         | sd          | 2.5%          | 25%           | 50%          |
|----|----------|--------------|-------------|---------------|---------------|--------------|
| ## | alpha    | 2.63088031   | 1.55351123  | -3.280620e-01 | 1.601987592   | 2.59596546   |
| ## | beta[1]  | 0.71458312   | 1.56044973  | -2.427518e+00 | -0.312851589  | 0.76773489   |
| ## | beta[2]  | 1.17547822   | 1.55728036  | -1.919101e+00 | 0.151485240   | 1.18804616   |
| ## | beta[3]  | 1.36591657   | 1.55764949  | -1.763078e+00 | 0.339267735   | 1.41456067   |
| ## | obs[1,1] | 0.08433790   | 0.03856245  | 2.828325e-02  | 0.056196420   | 0.07807455   |
| ## | obs[2,1] | 0.02571132   | 0.02456389  | 6.627765e-04  | 0.007296531   | 0.01879690   |
| ## | obs[1,2] | 0.46672002   | 0.08962580  | 3.111096e-01  | 0.403832389   | 0.45839242   |
| ## | obs[2,2] | 0.02227195   | 0.02208079  | 5.919903e-04  | 0.006317447   | 0.01554626   |
| ## | obs[1,3] | 0.43155662   | 0.09747926  | 2.150048e-01  | 0.372268451   | 0.44063263   |
| ## | obs[2,3] | 0.20783229   | 0.12894884  | 6.739447e-03  | 0.099809696   | 0.20003372   |
| ## | obs[1,4] | 0.01738545   | 0.01745395  | 5.270976e-04  | 0.005015423   | 0.01201634   |
| ## | obs[2,4] | 0.74418443   | 0.12566155  | 4.973153e-01  | 0.649203627   | 0.74769521   |
| ## | deviance | 237.85563026 | 10.12671074 | 2.144607e+02  | 231.810420342 | 239.74821795 |
| ## |          | 75%          | 97.5%       | Rhat          | n.eff         | overlap0     |
| ## | alpha    | 3.66354984   | 5.76943735  | 1.0032749     | 533           | 1 0.9566667  |
| ## | beta[1]  | 1.75929605   | 3.66213722  | 1.0023251     | 699           | 1 0.6800000  |
| ## | beta[2]  | 2.24282695   | 4.15940259  | 1.0024769     | 693           | 1 0.7729167  |
| ## | beta[3]  | 2.41962867   | 4.41745725  | 1.0025523     | 673           | 1 0.8112500  |
| ## | obs[1,1] | 0.10655996   | 0.17582138  | 1.0006356     | 1652          | 0 1.0000000  |
| ## | obs[2,1] | 0.03590389   | 0.09115278  | 0.9999976     | 2400          | 0 1.0000000  |
| ## | obs[1,2] | 0.52097363   | 0.66058873  | 1.0045805     | 448           | 0 1.0000000  |
| ## | obs[2,2] | 0.03097798   | 0.08135244  | 0.9999102     | 2400          | 0 1.0000000  |

```
## obs[1,3] 0.49992594 0.59799500 1.0059527 350 0 1.0000000
## obs[2,3] 0.30634204 0.46963470 1.0018176 923 0 1.0000000
## obs[1,4] 0.02390928 0.06355135 1.0026492 2400 0 1.0000000
## obs[2,4] 0.84608240 0.95335536 1.0016600 1005 0 1.0000000
## deviance 245.09298999 254.54832338 1.0016483 1317 0 1.0000000
```

```
par(mfrow = c(2,2))
hist(test$sims.list$alpha, breaks = 50, col = "gray", main = "intercept")
abline(v = 0, lty = 2)

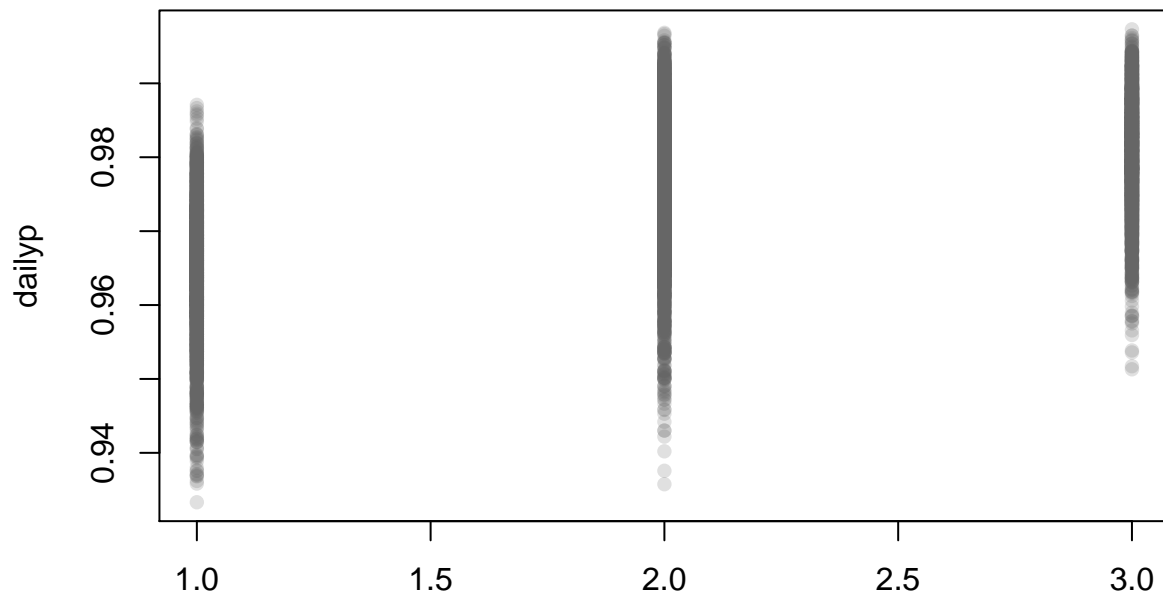
for(i in 1:3){
  hist(test$sims.list$beta[,i], breaks = 50, col = "gray", main = habitat[i])
  abline(v = 0, lty = 2)
}
```



So this tells us that savannahs are worse than average and woods are better than average. We could fit this without the intercept to get means instead of effects, if we wanted. From this we can also derive the probability of success in each habitat. Either daily survival probability:

```
dailyp = matrix(0, nrow = 3, ncol = dim(rhwo1$sims.list$beta)[1])
for(i in 1:3){
  dailyp[i,] = plogis(rhwo1$sims.list$alpha + rhwo1$sims.list$beta[,i])
}

matplot(dailyp, pch = 16, col = alpha("gray40", 0.2))
```



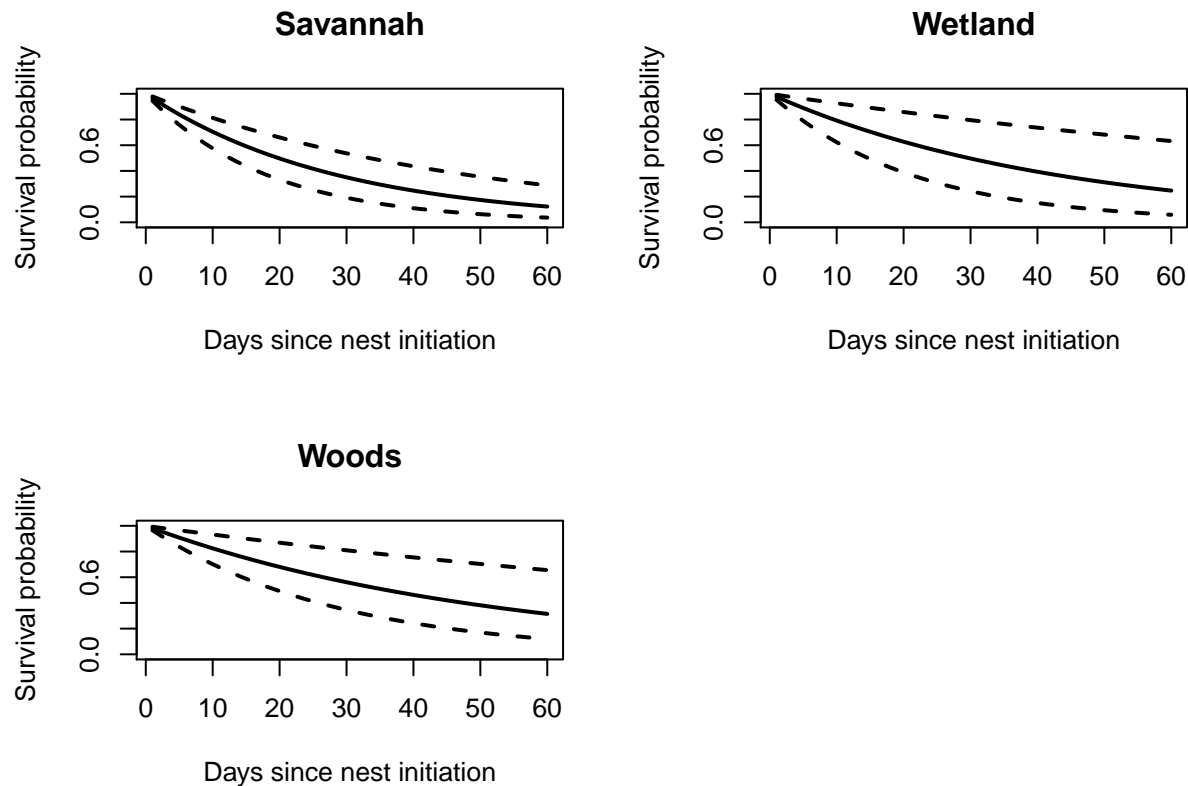
Or probability of surviving over time. Here the solid lines are the posterior means and dashed lines are the 95% credible intervals.

```
p = array(0, dim = c(3, 60, dim(rhwo1$sims.list$beta)[1]))
for(i in 1:3){
  for(t in 1:60){
    p[i,t,] = plogis(rhwo1$sims.list$alpha + rhwo1$sims.list$beta[,i])^t
  }
}

par(mfrow = c(2,2))
for(i in 1:3){
  medp = apply(p[i,,], 1, median)
  lp = apply(p[i,,], 1, function(x) quantile(x, probs = c(0.025)))
  up = apply(p[i,,], 1, function(x) quantile(x, probs = c(0.975)))

  plot(medp ~ c(1:60), type = "l", lwd = 2, xlab = "Days since nest initiation",
       ylab = "Survival probability", main = habitat[i], ylim = c(0,1))
  lines(lp, lty = 2, lwd = 2)
  lines(up, lty = 2, lwd = 2)
}
```





### Advantages and disadvantages

#### *Pros*

- \* can deal with uncertain fates explicitly
- \* very flexible approach to include any kind of random or fixed effects
- \* interpretation of results is the same as logistic regression, familiar to many ecologists

#### *Cons*

- \* model averaging/comparison less straightforward than AIC, but still possible! We can talk about how that would work
- \* still need to determine exposure time for each nest
- \* not the most precise estimates, but I think that's just a function of the data, not sure if we can avoid that

### Multievent model

I'm not going to go into a lot of detail here, but the third option is a multistate model with uncertainty in observations, ie a multievent model. I think that this is not necessary to answer the main research questions, and would only be useful if we decide these first two options won't work for one reason or another.