Code

* [Show All Code](#gjdgxs)
* [Hide All Code](#gjdgxs)
* [Download Rmd](#gjdgxs)

# Multi-species occupancy; Rota walkthrough. Christian Anderson 5/24/2023

Rota multi-species occupancy walkthrough

( <https://www.youtube.com/watch?v=tj_OCO77_sc> ) Christian Anderson 5/24/2023

Read in detection/non-detection data

#V1-V3 = weeks surveyed  
#rows = cameras  
bob <- read.csv('data/bobcat\_3wk.csv', header = F)  
coy <- read.csv('data/Coyote\_3wk.csv', header = F)  
fox <- read.csv('data/RedFox\_3wk.csv', header = F)  
  
head(fox, 10)

site level covariate data (covariates that only vary by site but stay constant for duration of survey)

* Dist\_5km = proportion disturbance within 5km
* HDens\_5km = housing density within 5km
* latitude / longitude = latitude and longitude (div by 100)
* People\_site = total people recorded / 1000
* trail = binary indicator of wheter camera was on trail (1) or not (0)

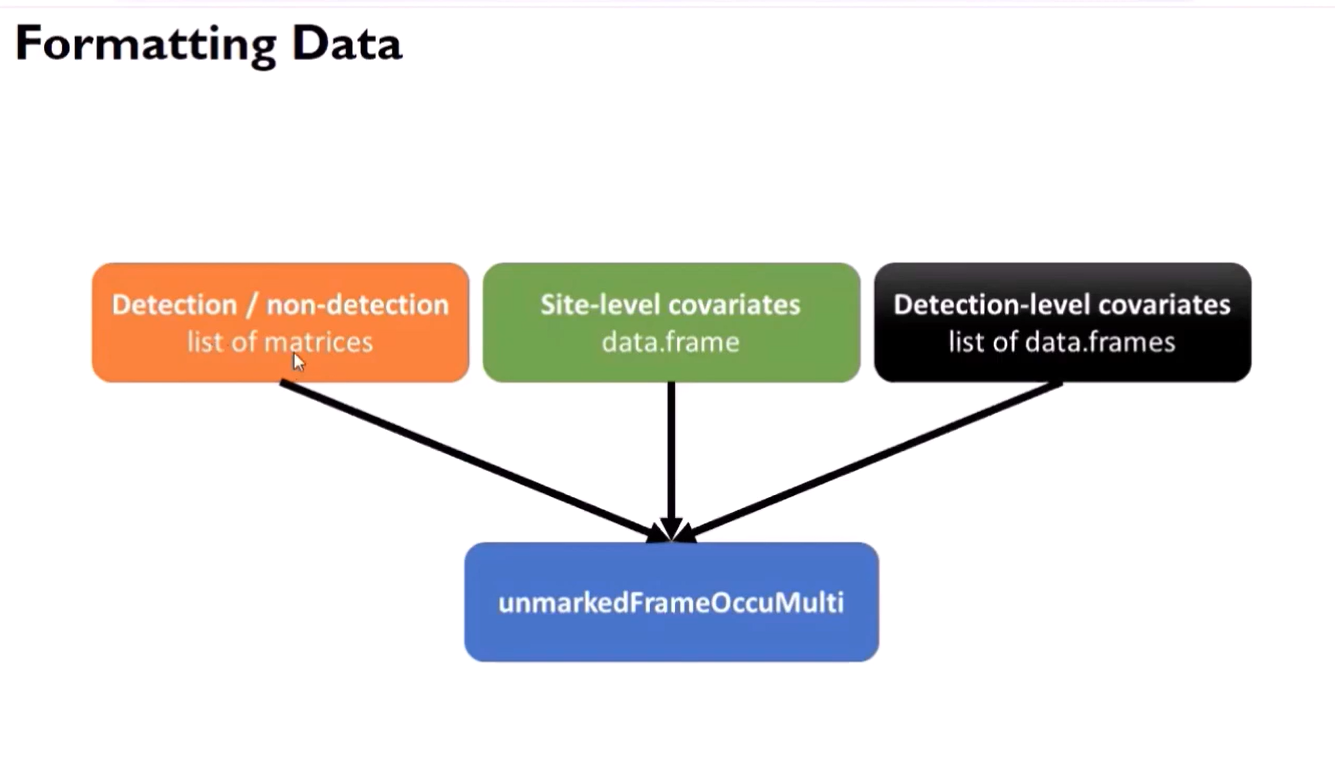
occ\_covs <- read.csv('data/psi\_cov3wk.csv')  
head(occ\_covs)

read in detection level covariate on column for every repliacte survey; average temperature over one week replicate period (example of covariate that changes from detection to detection)

det\_covs <- read.csv('data/detection data.csv')  
head(det\_covs)

apply unmarked package

library(unmarked)



place detection/non-detection into a named list

#each element of list must be a matrix  
y\_list <- list(bobcat = as.matrix(bob),  
 coyote = as.matrix(coy),  
 redfox = as.matrix(fox))

Place detection covariates into a named list

#each element of the list must be of the same dimension as each detection / non-detection mstrix  
 #same num of rows (each site) and columns(repl survey)  
  
det\_list <- list(temp= det\_covs)  
 #temperature is a covariate that can change based on when there is a instance of detection

combine data into an unmarkedFrameOccuMulti object:

#fn+f1 pulls up help menu  
msom\_data <- unmarkedFrameOccuMulti(y = y\_list,   
 siteCovs= occ\_covs,   
 obsCovs = det\_list)

Intercept only model, assuming independence

Syntax similar to other models

* need detection formula for each species
* need occupancy formula for each natural parameter included in model (3)
* need formulas (specified as character vector)
* need order of formulas follows detection / non-detection list (bob, coy, fox

Fit a Model

intercept only model that assumes independence among species. We do this by only allowing 1st order natural paramaters (maxOrder = 1) assumes species occur independently at each site

* essentially the same as fitting three single species occupancy models

fit\_1 <- occuMulti(detformulas = c('~1', '~1', '~1'),  
 stateformulas = c('~1', '~1', '~1'),  
 maxOrder = 1,  
 data = msom\_data)  
 #state formulas len=3 so three natural parameters, i,e. three species  
 #maxOrder = 1 makes it so that the model ignores natural params above first order> assumes species occur independently at each site

This is equivalent to fitting 3 single-species occupancy models

summary(fit\_1)

Intercept-only model, assuming dependence

* set maxOrder = 2 to estimate up to 2nd order nat. params
* permits dependence between species
* fixes all nat params > maxOrder at 0

fit\_2 <- occuMulti(detformulas = c('~1', '~1', '~1'),  
 stateformulas = c('~1', '~1', '~1',  
 '~1', '~1', '~1'),  
 maxOrder = 2,  
 data = msom\_data)  
 #State Formulas> there are three choose two different possiblities, one for each combo  
 #maxOrder = 2 sets natural parameters to 2nd order which allows for pairwise interactions between species

Output interpretation

(Estimate = slope coefficient )

* bobcat and coyote occur together more frequently than expected by chance (p<0.01)
* bobcat and red fox occur together less frequently than expected by chance (p<0.01)
* coyote and red fox occur together more frequently than expected by chance (p<0.01)
* AIC strongly favors the model incorporating dependence AIC= 6626.111 for fit\_2 and AIC= 6710.658 for fit\_1; it’s much smaller

Incorporating Covariates

* any param can be modeled as a function of covariates
* covariate modelsfor each param can be unique
* names of detection covariates correspond to names provided in named list
* names of occupancy covariates correspond to names in data.frame

fit\_3 <- occuMulti(detformulas = c('~temp', '~1', '~1'),  
 stateformulas = c( '~Dist\_5km', '~HDens\_5km', '~People\_site',  
 '~Latitude', '~1', '~1'),  
 maxOrder = 2,   
 data = msom\_data)

'Call:   
occuMulti(detformulas = c("~temp", "~1", "~1"), stateformulas = c("~Dist\_5km", "~HDens\_5km", "~People\_site", "~Latitude", "~1", "~1"), data = msom\_data, maxOrder = 2)   
  
Occupancy (logit-scale): Estimate SE z P(>|z|)   
[bobcat] (Intercept) -1.35970 0.19131 -7.11 1.18e-12   
[bobcat] Dist\_5km -22.95531 5.90074 -3.89 1.00e-04   
  
## ^log odds that a bobcat occurs at a location under the condition that all other species are absent  
  
[coyote] (Intercept) -1.48520 0.17195 -8.64 5.74e-18   
[coyote] HDens\_5km 0.00424 0.00232 1.83 6.74e-02   
[redfox] (Intercept) -2.45677 0.16226 -15.14 8.71e-52   
[redfox] People\_site 5.92772 1.27022 4.67 3.06e-06   
  
[bobcat:coyote] (Intercept) 8.89548 2.78768 3.19 1.42e-03   
[bobcat:coyote] Latitude -18.97578 7.30114 -2.60 9.35e-03   
## ^moving north, bobcat less likely occupy the same site as a coyote  
  
[bobcat:redfox] (Intercept) -1.55839 0.42126 -3.70 2.16e-04   
[coyote:redfox] (Intercept) 1.55320 0.26507 5.86 4.64e-09   
  
Detection (logit-scale): Estimate SE z P(>|z|) [bobcat] (Intercept) -1.248 0.1439 -8.67 4.16e-18   
[bobcat] temp -0.306 0.0848 -3.60 3.12e-04   
[coyote] (Intercept) -0.322 0.0756 -4.26 2.05e-05   
[redfox] (Intercept) -0.494 0.1295 -3.81 1.36e-04   
  
AIC: 6544.769   
Number of sites: 1437   
optim convergence code: 0   
optim iterations: 133   
Bootstrap iterations: 0

Sample interpretations:

* for every I-unit increase disturbance within 5km, the log odds of bobcat occurance, when *coyote and red fox are absent,* declines by 22.96
* When latitude = 0, bobcats are more likely to occur at a site when coyotes are present (and vice versa, p<0.01)
* For every I unit increase in Latitude, the log odds bobcats will occur at a site decreases by 18.98 if coyotes are also present, relative to equivalent sites where coyotes are absent
  + *i.e., as we move north, bobcats and coyotes are less likely to occupy the same site*
* For every I-unit increase in temperature, the log odds of bobcat detection decreases by 0.31

Conditional occupancy probability

Calculation of conditional and marginal occupancy probabilities is done with predict function

* create a data.frame for predictions
  + the procedure is equivalent to creating data frames for all other applications of predict
* Include complete range of observed latitude; hold all other variables at their mean

nd\_cond <- data.frame(  
 Dist\_5km = rep(mean(occ\_covs$Dist\_5km), 100),   
 HDens\_5km = rep(mean(occ\_covs$HDens\_5km), 100),   
 People\_site = rep(mean(occ\_covs$People\_site), 100),   
 Latitude = seq(min(occ\_covs$Latitude), max(occ\_covs$Latitude),  
 length.out = 100)  
)

Conditional occupancy probability

Predicting bobcat occurance when coyotes are present

* <species> indicates which species we assume when predicting occupancy
* <cond> indicateswhich species we are assuming is present or absent

```r  
bob\_coy\_1 <- predict(fit\_3, type = 'state' ,species = 'bobcat',   
 cond = 'coyote', newdata = nd\_cond)  
 #what is the probability that a bobcat will be at a site on the condition that a coyote was present at that site  
   
## Bootstrapping confidence intervals with 100 samples  
  
```

Predicting bobcat occurrence when coyotes are absent

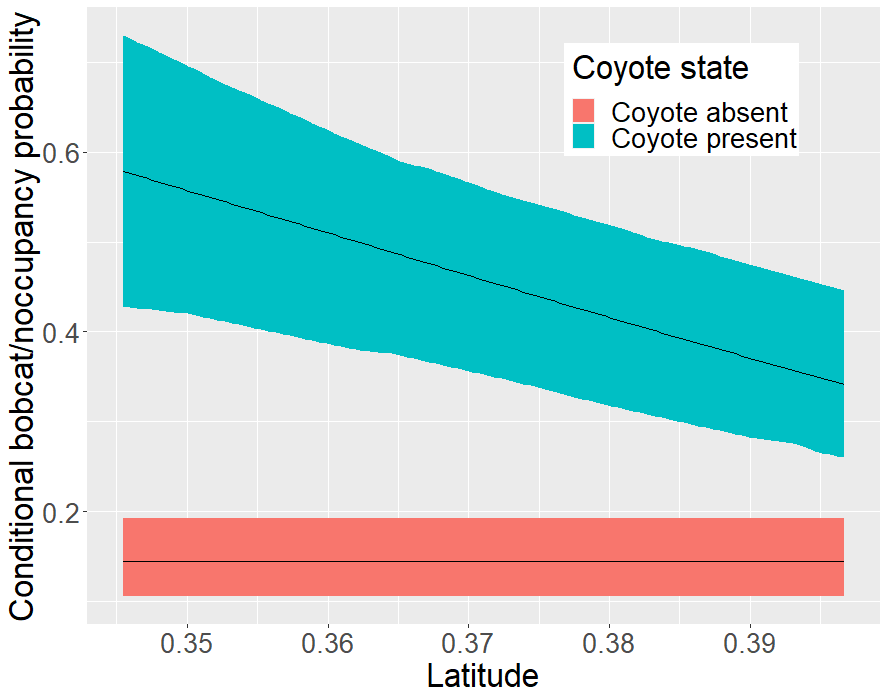
* putting a - in front of coyote tells *predict* you wish to assume coyotes are absent

bob\_coy\_0 <- predict(fit\_3, type = 'state', species = 'bobcat',   
 cond = '-coyote', newdata = nd\_cond)  
 #condition on coyote being absent  
  
## Bootstrapping confidence intervals with 100 samples

Plotting

gg\_df\_cond <- data.frame(  
 latitude = rep(nd\_cond$Latitude, 2),   
 occupancy = c(bob\_coy\_1$Predicted,   
 bob\_coy\_0$Predicted),  
 low = c(bob\_coy\_1$lower,  
 bob\_coy\_0$lower),  
 high = c(bob\_coy\_1$upper,  
 bob\_coy\_0$upper),  
 conditional = rep(c('Coyote present', 'Coyote absent'),  
 each = 100)  
)

library(ggplot2)  
  
cond\_fig <- ggplot(gg\_df\_cond, aes(x= latitude, y= occupancy,   
 group = conditional)) +  
 geom\_ribbon(aes(ymin = low, ymax = high, fill = conditional)) +   
 geom\_line() +  
 ylab('Conditional bobcat/noccupancy probability') +  
 xlab('Latitude') +   
 labs(fill = 'Coyote state') +   
 theme(text = element\_text(size = 25),   
 legend.position = c(0.75, 0.85))



Marginal occupancy probability

nd\_marg <- data.frame(  
 Dist\_5km = seq(min(occ\_covs$Dist\_5km), max(occ\_covs$Dist\_5km),  
 length.out = 100),  
 HDens\_5km = rep(mean(occ\_covs$HDens\_5km), 100),   
 People\_site = rep(mean(occ\_covs$People\_site), 100),   
 Latitude = rep(mean(occ\_covs$Latitude), 100)  
)

omitting the conditional argument to calculate marginal occupancy probability

bob\_marg <- predict(fit\_3, type = 'state', species = 'bobcat',   
 newdata = nd\_marg)

figure

gg\_df\_marg <- data.frame(  
 hd = nd\_marg$Dist\_5km,   
 occupancy = bob\_marg$Predicted,   
 low = bob\_marg$lower,   
 high = bob\_marg$upper  
   
)

marg\_fig <- ggplot(gg\_df\_marg, aes(x = hd, y= occupancy)) +  
 geom\_ribbon(aes(ymin = low, ymax = high), alpha = .5) +  
 geom\_line() +   
 ylab('Marginal bobcat\noccupancy probability') +  
 theme(text = element\_text(size = 25))

