

# Exploratory N-Mixture/CMR - Pennsylvania

## Load Packages

```
if (!require(librarian)){
  install.packages("librarian")
  library(librarian)
}

librarian::shelf(tidyverse, RPostgres, DBI, unmarked, here, lubridate, kableExtra)
```

## Connect to survey\_data schema in ribbitr database

```
tryCatch({
  drv <- dbDriver("Postgres")
  print("Connecting to Database...")
  connection <- dbConnect(drv,
    dbname = Sys.getenv("aws_dbname"),
    host = Sys.getenv("aws_host"),
    port = Sys.getenv("aws_port"),
    user = Sys.getenv("aws_user"),
    password = Sys.getenv("aws_password"),
    timezone=NULL)
  print("Database Connected!")
},
error=function(cond) {
  print("Unable to connect to Database.")
})

#search path
dbExecute(connection, "set search_path to survey_data")
```

## Query 2022 N-Mix Penn data

```
# Data
nmix_q <- "select r.region, s.site, v.date, v.survey_time, s2.duration_minutes,
              c.species_capture, c.capture_type
from region r
join site s on r.region_id = s.region_id
full join visit v on s.site_id = v.site_id
join survey s2 on v.visit_id = s2.visit_id
join capture c on s2.survey_id = c.survey_id
where r.region = 'pennsylvania'
and v.date > '2022-01-01';"

nmix_raw_data <- dbGetQuery(connection, nmix_q) %>%
  select(!c(region, survey_time, duration_minutes)) %>%
  arrange(date)

#write_csv(nmix_raw_data, here("data", "nmix_raw_data.csv"))

# find all visits
visit_nmix_q <- "select r.region, s.site, v.date, v.survey_time
from region r
join site s on r.region_id = s.region_id
join visit v on s.site_id = v.site_id
where r.region = 'pennsylvania'
and v.date > '2022-01-01';"

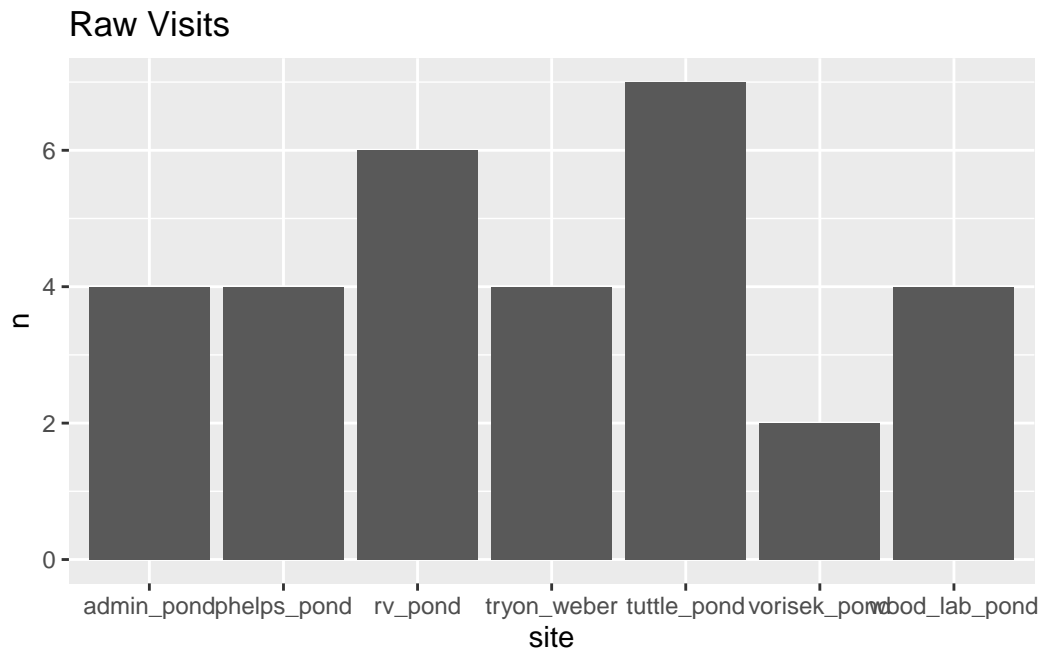
nmix_raw_visits <-dbGetQuery(connection, visit_nmix_q) %>%
  arrange(date) %>%
  select(site, date)

#write_csv(nmix_raw_visits, here("data", "nmix_raw_visits.csv"))
```

## Visualize all Visits

```
#nmix_raw_visits <- read_csv(here("data", "nmix_raw_visits.csv"))  
#nmix_raw_data <- read_csv(here("data", "nmix_raw_data.csv"))
```

```
viz <- nmix_raw_visits %>%  
  group_by(site) %>%  
  summarise(n = n())  
  
ggplot(data = viz) +  
  geom_col(aes(x=site, y = n)) +  
  ggtitle("Raw Visits")
```



## Clean up species list with counts

Filter for CMR focal species and summarize species counts. Then populate zeros for 3 focal species into the data set for visits when captures did not occur.

```
# filter for CMR focal species and summarize counts
n_mix_mid_clean_up <- nmix_raw_data %>%
  filter(species_capture %in% c("pseudacris_crucifer", "rana_catesbeiana",
                                "rana_clamitans")) %>%
  mutate(capture_type = if_else(is.na(capture_type), "new", capture_type)) %>%
  group_by(date, site, species_capture) %>%
  summarise(n = n()) %>%
  ungroup()

# populate zeros
nmix_clean_up <- nmix_raw_visits %>%
  left_join(n_mix_mid_clean_up) %>%
  complete(nesting(date, site),
           species_capture = unique(n_mix_mid_clean_up$species_capture),
           fill = list(n = 0))

kable(head(nmix_clean_up, n = 15))
```

date	site	species_capture	n
2022-05-04	wood_lab_pond	pseudacris_crucifer	2
2022-05-04	wood_lab_pond	rana_catesbeiana	2
2022-05-04	wood_lab_pond	rana_clamitans	1
2022-05-05	phelps_pond	pseudacris_crucifer	4
2022-05-05	phelps_pond	rana_catesbeiana	4
2022-05-05	phelps_pond	rana_clamitans	7
2022-05-11	admin_pond	pseudacris_crucifer	4
2022-05-11	admin_pond	rana_catesbeiana	9
2022-05-11	admin_pond	rana_clamitans	5
2022-05-12	rv_pond	pseudacris_crucifer	16
2022-05-12	rv_pond	rana_catesbeiana	1
2022-05-12	rv_pond	rana_clamitans	0
2022-05-18	admin_pond	pseudacris_crucifer	2
2022-05-18	admin_pond	rana_catesbeiana	15
2022-05-18	admin_pond	rana_clamitans	3

## rana\_catesbeiana

### N-Mixture Table formatting

Filter data for rana\_catesbeiana, tally the number of visits, pivot data frame into correct matrix form, and finally populate zeros into the NAs if sites were visited.

```
bull_frog_visits <- nmix_clean_up %>%
  select(site, date, species_capture, n) %>%
  filter(species_capture == "rana_catesbeiana") %>%
  select(!species_capture) %>%
  group_by(site) %>%
  mutate(n_visit = match(date, unique(date)),
         n_visit = paste0("visit_", n_visit, sep = "")) %>%
  select(!date) %>%
  ungroup() %>%
  pivot_wider(names_from = c("n_visit"), values_from = c("n")) %>%
  group_by(site) %>%
  mutate(across(contains("visit"),
               ~ifelse(is.na(.) &
                      !is.na(lag(.)), 0, .)))

kable(bull_frog_visits)
```

site	visit_1	visit_2	visit_3	visit_4	visit_5	visit_6	visit_7
wood_lab_pond	2	4	3	0	NA	NA	NA
phelps_pond	4	3	6	0	NA	NA	NA
admin_pond	9	15	20	0	NA	NA	NA
rv_pond	1	7	5	13	29	11	NA
tuttle_pond	15	22	0	39	31	29	20
tryon_weber	0	0	0	NA	NA	NA	NA
vorisek_pond	14	13	NA	NA	NA	NA	NA

### static n-mixture models no covariates

```
bull_frog_unmarked_nmixture <- bull_frog_visits %>%
  ungroup() %>%
  select(!site)
```

```
umf <- unmarkedFramePCount(y = bull_frog_unmarked_nmixture)

summary(umf)
```

unmarkedFrame Object

7 sites

Maximum number of observations per site: 7

Mean number of observations per site: 4.29

Sites with at least one detection: 6

Tabulation of y observations:

0	1	2	3	4	5	6	7	9	11	13	14	15	20	22	29
7	1	1	2	2	1	1	1	1	1	2	1	2	2	1	2
31	39	<NA>													
1	1	19													

```
fm1 <- pcount( ~ 1 ~ 1,
               data = umf,
               engine = "R")
```

Warning in pcount(~1 ~ 1, data = umf, engine = "R"): K was not specified and was set to 139.

```
summary(fm1)
```

Call:

```
pcount(formula = ~1 ~ 1, data = umf, engine = "R")
```

Abundance (log-scale):

Estimate	SE	z	P(> z )
3.71	0.196	18.9	6.13e-80

Detection (logit-scale):

Estimate	SE	z	P(> z )
-1.2	0.236	-5.06	4.14e-07

AIC: 401.9176  
Number of sites: 7  
optim convergence code: 0  
optim iterations: 24  
Bootstrap iterations: 0

```
backTransform(fm1, "state")
```

Backtransformed linear combination(s) of Abundance estimate(s)

Estimate	SE	LinComb	(Intercept)
40.7	7.97	3.71	1

Transformation: exp

```
backTransform(fm1, "det")
```

Backtransformed linear combination(s) of Detection estimate(s)

Estimate	SE	LinComb	(Intercept)
0.232	0.0421	-1.2	1

Transformation: logistic

**Assuming the sites and enviro variables are exactly the same we can say there is 40.7 +-7.9 frogs at each site with 23 % chance of detecting each individual.**

## pseudacris\_crucifer

### N-Mixture Table formatting

Filter data for `pseudacris_crucifer`, tally the number of visits, pivot data frame into correct matrix form, and finally populate zeros into the NAs if sites were visited.

```
peep_frog_visits <- nmix_clean_up %>%
  select(site, date, species_capture, n) %>%
  filter(species_capture == "pseudacris_crucifer") %>%
  select(!species_capture) %>%
  group_by(site) %>%
  mutate(n_visit = match(date, unique(date)),
         n_visit = paste0("visit_", n_visit, sep = "")) %>%
  select(!date) %>%
  ungroup() %>%
  pivot_wider(names_from = c("n_visit"), values_from = c("n")) %>%
  group_by(site) %>%
  mutate(across(contains("visit"),
               ~ifelse(is.na(.) &
                       !is.na(lag(.)), 0, .)))

kable(peep_frog_visits)
```

site	visit_1	visit_2	visit_3	visit_4	visit_5	visit_6	visit_7
wood_lab_pond	2	0	0	0	NA	NA	NA
phelps_pond	4	0	0	0	NA	NA	NA
admin_pond	4	2	3	1	NA	NA	NA
rv_pond	16	0	0	0	0	0	NA
tuttle_pond	0	0	0	0	0	0	0
tryon_weber	0	0	0	NA	NA	NA	NA
vorisek_pond	0	0	NA	NA	NA	NA	NA



## static n-mixture model no covariates

```
peep_unmarked_nmixture <- peep_frog_visits %>%
  ungroup() %>%
  #filter(capture_type == "new") %>%
  select(!c(site))

umf <- unmarkedFramePCount(y = peep_unmarked_nmixture)

summary(umf)
```

unmarkedFrame Object

7 sites  
Maximum number of observations per site: 7  
Mean number of observations per site: 4.29  
Sites with at least one detection: 4

Tabulation of y observations:

0	1	2	3	4	16	<NA>
23	1	2	1	2	1	19

```
fm1 <- pcount(~1 ~1, data = umf)
```

Warning in pcount(~1 ~ 1, data = umf): K was not specified and was set to 116.

```
backTransform(fm1, "state")
```

Backtransformed linear combination(s) of Abundance estimate(s)

Estimate	SE	LinComb	(Intercept)
85.8	24.4	4.45	1

Transformation: exp

```
backTransform(fm1, "det")
```

Backtransformed linear combination(s) of Detection estimate(s)

Estimate	SE	LinComb	(Intercept)
0.0124	0.00411	-4.38	1

Transformation: logistic

**Assuming the sites and enviro variables are exactly the same we can estimate there is 85.7  
+-24.4 frogs at each site with a 1.2% chance of detecting each individual.**

## rana\_clamitans

### N-Mixture Table Formatting

Filter data for rana\_clamitans, tally the number of visits, pivot data frame into correct matrix form, and finally populate zeros into the NAs if sites were visited.

```
green_frog_visits <- nmix_clean_up %>%
  select(site, date, species_capture, n) %>%
  filter(species_capture == "rana_clamitans") %>%
  #select(!species_capture) %>%
  group_by(site) %>%
  mutate(n_visit = match(date, unique(date)),
         n_visit = paste0("visit_", n_visit, sep = "")) %>%
  select(!date) %>%
  ungroup() %>%
  group_by(site, n_visit, ) %>%
  summarise(n = sum(n)) %>%
  ungroup() %>%
  pivot_wider(names_from = c("n_visit"), values_from = c("n")) %>%
  #add_row(site = "phelps_pond", capture_type = "recapture") %>%
  group_by(site) %>%
  mutate(across(contains("visit"),
                ~ifelse(is.na(.) &
                        !is.na(lag(.)), 0, .)))
```

`summarise()` has grouped output by 'site'. You can override using the  
`.groups` argument.

```
kable(green_frog_visits)
```

site	visit_1	visit_2	visit_3	visit_4	visit_5	visit_6	visit_7
admin_pond	5	3	0	0	NA	NA	NA
phelps_pond	7	10	16	0	NA	NA	NA
rv_pond	0	2	1	7	3	0	NA
tryon_weber	0	2	2	NA	NA	NA	NA
tuttle_pond	19	3	0	3	2	1	2
vorisek_pond	8	1	NA	NA	NA	NA	NA
wood_lab_pond	1	22	3	0	NA	NA	NA

### static n-mixture with no co-variates

```
green_frog_unmarked_nmixture <- green_frog_visits %>%
  ungroup() %>%
  select(!c(site))

umf <- unmarkedFramePCount(y = green_frog_unmarked_nmixture)

summary(umf)
```

#### unmarkedFrame Object

7 sites

Maximum number of observations per site: 7

Mean number of observations per site: 4.29

Sites with at least one detection: 7

Tabulation of y observations:

0	1	2	3	5	7	8	10	16	19	22	<NA>
8	4	5	5	1	2	1	1	1	1	1	19

```
fm1 <- pcount(~1 ~1, data = umf)
```

Warning in pcount(~1 ~ 1, data = umf): K was not specified and was set to 122.

```
backTransform(fm1, "state")
```

Backtransformed linear combination(s) of Abundance estimate(s)

Estimate	SE	LinComb	(Intercept)
94.8	13.1	4.55	1

Transformation: exp

```
backTransform(fm1, "det")
```

Backtransformed linear combination(s) of Detection estimate(s)

Estimate	SE	LinComb (Intercept)
0.0434	0.00693	-3.09 1

Transformation: logistic

**Assuming the sites and enviro variables are exactly the same we estimate there is 94.8  $\pm$  13.1 frogs at each sites and with 4.3% of detecting each individual.**

## Query CMR data

```
# Data
cmr_q <- "select r.region, s.site, v.date, v.survey_time, s2.duration_minutes,
  c.species_capture, c.capture_type, cmr.capture_date, cmr.cmr_id
  from region r
  join site s on r.region_id = s.region_id
  full join visit v on s.site_id = v.site_id
  join survey s2 on v.visit_id = s2.visit_id
  join capture c on s2.survey_id = c.survey_id
  join penn_cmr cmr on c.capture_mark_recapture = cmr.capture_mark_recapture
  where r.region = 'pennsylvania'
  and v.date > '2022-01-01';"

cmr_raw_data <- dbGetQuery(connection, cmr_q) %>%
  select(!c(region, survey_time, duration_minutes)) %>%
  arrange(date)

#write_csv(cmr_raw_data, here("data", "cmr_raw_data.csv"))

# find all visits
visit_cmr_q <- "select r.region, s.site, v.date, v.survey_time
  from region r
  join site s on r.region_id = s.region_id
  join visit v on s.site_id = v.site_id
  where r.region = 'pennsylvania'
  and v.date > '2022-01-01';"

cmr_raw_visits <-dbGetQuery(connection, visit_cmr_q) %>%
  arrange(date) %>%
  select(site, date)

#write_csv(cmr_raw_visits, here("data", "cmr_raw_visits.csv"))
```

## Bullfrog CMR data

### Matrix: individual counts by visits

```
#cmr_raw_data <- read_csv(here("data", "cmr_raw_data.csv"))
#mr_raw_visits <- read_csv(here("data", "cmr_raw_visits.csv"))

bull_mid_clean_up <- cmr_raw_data %>%
  filter(species_capture == "rana_catesbeiana") %>%
  select(!c(capture_date, capture_type)) %>%
  unite(species_capture, c(species_capture, cmr_id), sep = "_") %>%
  unique()

bull_pop_zeros <- cmr_raw_visits %>%
  left_join(bull_mid_clean_up) %>%
  complete(nesting(date, site),
           fill = list(n = 0)) %>%
  ungroup() %>%
  group_by(site) %>%
  mutate(n_visit = match(date, unique(date)),
         n_visit = paste0("visit_", n_visit, sep = "")) %>%
  select(!date) %>%
  ungroup()
```

Joining with `by = join\_by(site, date)`

```
clean_bull <- bull_pop_zeros %>%
  group_by(site, species_capture, n_visit) %>%
  reframe(n = n()) %>%
  mutate(n = if_else(is.na(species_capture), NA, n),
         n = if_else(n == 2, 1, n)) %>%
  pivot_wider(names_from = c("n_visit"), values_from = c("n")) %>%
  relocate(visit_1, .before = visit_2) %>%
  relocate(visit_3, .after = visit_2) %>%
  drop_na(species_capture) %>%
  filter(!site == "tryon_weber") %>%
  select(!site) %>%
  mutate_all(~replace_na(.,0))
```

## Final CMR matrix

```
clean_bull$captureHistory <- paste(clean_bull$visit_1, clean_bull$visit_2, clean_bull$visit_3,
                                   clean_bull$visit_4, clean_bull$visit_5, clean_bull$visit_6,
                                   clean_bull$visit_7,
                                   sep = "")

clean_bull <- clean_bull %>%
  ungroup() %>%
  group_by(species_capture, captureHistory) %>%
  unique()

lev <- unique(clean_bull$captureHistory)

clean_bull$captureHistory <- factor(clean_bull$captureHistory, levels = lev)

bull_table <- table(clean_bull$species_capture, clean_bull$captureHistory)

kable(head(bull_table, n = 15))
```

	0010000	0110000	0100000	0110010	0010010	0001100	0001000	0000000
rana_catesbeiana_A1A2	1	0	0	0	0	0	0	0
rana_catesbeiana_A2	0	1	1	1	0	0	0	0
rana_catesbeiana_A2A3	0	1	0	0	1	0	0	0
rana_catesbeiana_A2A3A4	0	0	0	0	0	0	0	0
rana_catesbeiana_A2A3B2	0	0	0	0	0	1	0	0
rana_catesbeiana_A2A3B2B3	0	0	0	0	0	0	1	0
rana_catesbeiana_A2A3B2B4	0	0	0	0	0	1	0	0
rana_catesbeiana_A2A3B3	0	0	0	0	0	0	1	0
rana_catesbeiana_A2A3B4	0	0	0	0	0	0	1	0
rana_catesbeiana_A2A4	0	1	0	0	1	0	0	0
rana_catesbeiana_A2A4B2	0	0	0	0	0	0	1	0
rana_catesbeiana_A2A4B3	0	0	0	0	0	0	1	0
rana_catesbeiana_A2A4B4	0	0	0	0	0	0	1	0
rana_catesbeiana_A2B2	1	0	1	0	1	0	0	0
rana_catesbeiana_A2B2B3	0	0	0	0	0	0	1	0



## obs covariates, piFun equal detection, Model runn

```

class(bull_table) <- "matrix"

o2y <- matrix(1, 7, 19)

crPiFun <- function(p) {
  p1 <- p[,1]
  p2 <- p[,2]
  p3 <- p[,3]
  p4 <- p[,4]
  p5 <- p[,5]
  p6 <- p[,6]
  p7 <- p[,7]
  cbind("0010000" = (1-p1) * (1-p2) * p3 * (1-p4) * (1-p5) * (1-p6)* (1-p7), #1: 0010
        "0110000" = (1-p1) * p2 * p3 * (1-p4) * (1-p5) * (1-p6)* (1-p7), #2: 0110
        "0100000" = (1-p1) * p2 * (1-p3) * (1-p4) * (1-p5) * (1-p6)* (1-p7), #3: 0100
        "0110010" = (1-p1) * p2 * p3 * (1-p4) * (1-p5) * p6* (1-p7), #4: 0110
        "0010010" = (1-p1) * (1-p2) * p3 * (1-p4) * (1-p5) * p6 * (1-p7), #5: 0010
        "0001100" = (1-p1) * (1-p2) * (1-p3) * p4 * p5 * (1-p6) * (1-p7), #6: 0001
        "0001000" = (1-p1) * (1-p2) * (1-p3) * p4 * (1-p5) * (1-p6)* (1-p7), #7: 0001
        "0001010" = (1-p1) * (1-p2) * (1-p3) * p4 * (1-p5) * p6 * (1-p7), #8: 0001
        "0100010" = (1-p1) * p2 * (1-p3) * (1-p4) * (1-p5) * p6 * (1-p7), #9: 0100
        "1101010" = p1 * p2 * (1-p3) * p4 * (1-p5) * p6 * (1-p7), #10: 110
        "1101100" = p1 * p2 * (1-p3) * p4 * p5 * (1-p6)* (1-p7), #11: 110
        "0101100" = (1-p1) * p2 * (1-p3) * p4 * p5 * (1-p6)* (1-p7), #12: 010
        "1001000" = p1 * (1-p2) * (1-p3) * p4 * (1-p5) * (1-p6) * (1-p7), #13: 100
        "1101000" = p1 * p2 * (1-p3) * p4 * (1-p5) * (1-p6)* (1-p7), #14: 110
        "0101000" = (1-p1) * p2 * (1-p3) * p4 * (1-p5) * (1-p6)* (1-p7), #15: 010
        "0101001" = (1-p1) * p2 * (1-p3) * p4 * (1-p5) * (1-p6)* p7, #16: 010
        "1001010" = p1 * (1-p2) * (1-p3) * p4 * (1-p5) * p6 * (1-p7), #17: 100
        "1100000" = p1 * p2 * (1-p3) * (1-p4) * (1-p5) * (1-p6)* (1-p7), #18: 110
        "1000000" = p1 * (1-p2) * (1-p3) * (1-p4) * (1-p5) * (1-p6) * (1-p7) #19: 100
  )
}

umf <- unmarkedFrameMPois(y = bull_table, piFun = "crPiFun", obsToY = o2y)

fm <- multinomPois(~1 ~1, umf, engine = "R")

```

```
backTransform(fm, "state")
```

Backtransformed linear combination(s) of Abundance estimate(s)

Estimate	SE	LinComb	(Intercept)	
7.76	0.839	2.05		1

Transformation: exp

```
backTransform(fm, "det")
```

Backtransformed linear combination(s) of Detection estimate(s)

Estimate	SE	LinComb	(Intercept)	
0.3	0.0266	-0.846		1

Transformation: logistic

**Assuming sites are identical we estimate there are 8 individuals at each site? With a 30% chance of detecting each individual? Not fully sure how to interpret results.**