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

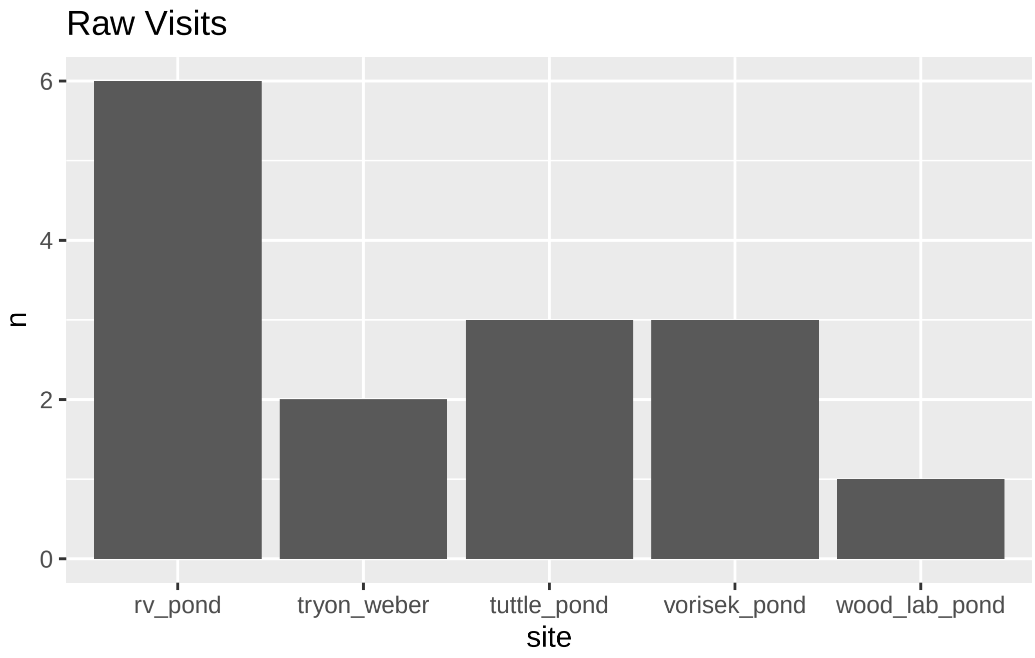
```
tryCatch({
    drv <- dbDriver("Postgres")
    print("Connecting to Database…")
    connection <- dbConnect(drv,
                 dbname = Sys.getenv("aws_dbname"),
                 host = Sys.getenv("aws_host"),
                 port = Sys.getenv("aws_port"),
                 user = Sys.getenv("aws_user"),
                 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,
          v2.species_ves, v2.count
          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 ves v2 on s2.survey_id = v2.survey_id
          where r.region = 'pennsylvania'
          and v.date > '2023-01-01';"

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, s2.detection_type, s2.obs
                from region r
                join site s on r.region_id = s.region_id
                join visit v on s.site_id = v.site_id
                join survey s2 on v.visit_id = s2.visit_id
                where r.region = 'pennsylvania'
                and v.date > '2023-01-01'
                and s2.detection_type = 'visual';"

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 VES 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_ves %in% c("pseudacris_crucifer", "rana_catesbeiana",
                                "rana_clamitans", "rana_pipiens")) %>%
  #mutate(capture_type = if_else(is.na(capture_type), "new", capture_type)) %>%
  group_by(date, site, species_ves) %>%
  summarise(n = sum(count)) %>%
  ungroup()

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

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

date	site	species_ves	n
2023-05-16	wood_lab_pond	rana_catesbeiana	0
2023-05-16	wood_lab_pond	rana_clamitans	4
2023-05-22	rv_pond	rana_catesbeiana	6
2023-05-22	rv_pond	rana_clamitans	7
2023-05-23	rv_pond	rana_catesbeiana	6
2023-05-23	rv_pond	rana_clamitans	19
2023-05-24	rv_pond	rana_catesbeiana	8
2023-05-24	rv_pond	rana_clamitans	20
2023-05-29	tuttle_pond	rana_catesbeiana	14
2023-05-29	tuttle_pond	rana_clamitans	13
2023-05-30	tuttle_pond	rana_catesbeiana	19
2023-05-30	tuttle_pond	rana_clamitans	10
2023-05-30	vorisek_pond	rana_catesbeiana	10
2023-05-30	vorisek_pond	rana_clamitans	2
2023-05-31	tuttle_pond	rana_catesbeiana	22

rana_catesbeiana

N-Mixture Table formatting

Filter data for `rana_catesbeiana`, tally the numbner 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_ves, n) %>%
  filter(species_ves == "rana_catesbeiana") %>%
  select(!species_ves) %>%
  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
wood_lab_pond	0	NA	NA	NA	NA	NA
rv_pond	6	6	8	12	12	10
tuttle_pond	14	19	22	NA	NA	NA
vorisek_pond	10	10	13	NA	NA	NA

Bullfrog static n-mixture models no covariates - 3 repeat sites

```
first_bull_frog_unmarked_nmixture <- bull_frog_visits %>%
  ungroup() %>%
  filter(!site == "wood_lab_pond") %>%
  select(!c(site, c(visit_4:visit_6)))

second_bull_frog_unmarked_nxmixture <- bull_frog_visits %>%
```

```r
    ungroup() %>%
    select(site, c(visit_4:visit_6)) %>%
    slice(2) %>%
    rename(visit_1 = visit_4,
           visit_2 = visit_5,
           visit_3 = visit_6) %>%
    select(!c(site))

bull_frog_nmixture_3_visits <- rbind(first_bull_frog_unmarked_nmixture, second_bull_frog_u

bf_umf_3 <- unmarkedFramePCount(y = bull_frog_nmixture_3_visits)

summary(bf_umf_3)
```

```
unmarkedFrame Object

4 sites
Maximum number of observations per site: 3
Mean number of observations per site: 3
Sites with at least one detection: 4

Tabulation of y observations:
 6  8 10 12 13 14 19 22
 2  1  3  2  1  1  1  1
```

```r
fm1_bf3 <- pcount( ~ 1 ~ 1,
               data = bf_umf_3,
               engine = "R")
```

```
Warning in pcount(~1 ~ 1, data = bf_umf_3, engine = "R"): K was not specified
and was set to 122.
```

```r
summary(fm1_bf3)
```

```


Call:
pcount(formula = ~1 ~ 1, data = bf_umf_3, engine = "R")

Abundance (log-scale):
 Estimate    SE    z  P(>|z|)
     2.89 0.248 11.7 2.05e-31

Detection (logit-scale):
 Estimate    SE     z P(>|z|)
    0.638 0.649 0.984   0.325

AIC: 66.52254
Number of sites: 4
optim convergence code: 0
optim iterations: 23
Bootstrap iterations: 0
```

```
backTransform(fm1_bf3, "state")
```

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

 Estimate   SE LinComb (Intercept)
     18.1 4.49    2.89           1

Transformation: exp
```

```
backTransform(fm1_bf3, "det")
```

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

 Estimate    SE LinComb (Intercept)
    0.654 0.147   0.638           1

Transformation: logistic
```

Assuming the sites and enviro variables are exactly the same we can say there is 18.08 +-4.48 frogs at each site with 65 % chance of detecting each individual when 3 repeat visits occur nightly.

Bullfrog static n-mixture models no covariates - 2 repeat sites

```
first_bull_frog_unmarked_nmixture <- bull_frog_visits %>%
  ungroup() %>%
  filter(!site == "wood_lab_pond") %>%
  select(!c(site, c(visit_3:visit_6)))

second_bull_frog_unmarked_nxmixture <- bull_frog_visits %>%
  ungroup() %>%
  select(site, c(visit_4:visit_5)) %>%
  slice(2) %>%
  rename(visit_1 = visit_4,
         visit_2 = visit_5) %>%
  select(!c(site))

bull_frog_nmixture_2_visits <- rbind(first_bull_frog_unmarked_nmixture, second_bull_frog_u

bf_umf_2 <- unmarkedFramePCount(y = bull_frog_nmixture_2_visits)

summary(bf_umf_2)
```

```
unmarkedFrame Object

4 sites
Maximum number of observations per site: 2
Mean number of observations per site: 2
Sites with at least one detection: 4

Tabulation of y observations:
 6 10 12 14 19
 2  2  2  1  1
```

```
fm1_bf2 <- pcount( ~ 1 ~ 1,
               data = bf_umf_2,
               engine = "R")
```

```
Warning in pcount(~1 ~ 1, data = bf_umf_2, engine = "R"): K was not specified
and was set to 119.
```

```
summary(fm1_bf2)
```

```
Call:
pcount(formula = ~1 ~ 1, data = bf_umf_2, engine = "R")

Abundance (log-scale):
 Estimate    SE    z  P(>|z|)
     2.63 0.229 11.5 1.48e-30

Detection (logit-scale):
 Estimate    SE    z P(>|z|)
     1.38 0.956 1.45   0.148

AIC: 43.84742
Number of sites: 4
optim convergence code: 0
optim iterations: 21
Bootstrap iterations: 0
```

```
backTransform(fm1_bf2, "state")
```

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

 Estimate   SE LinComb (Intercept)
     13.9 3.19    2.63           1

Transformation: exp
```

```
backTransform(fm1_bf2, "det")
```

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

 Estimate    SE LinComb (Intercept)
    0.799 0.153    1.38           1

Transformation: logistic
```

Assuming the sites and enviro variables are exactly the same we can say there is 13.91 +-3.18 frogs at each site with 79 % chance of detecting each individual when 2 repeat visits occur nightly.

rana_clamitans

N-Mixture Table Formatting

Filter data for `rana_clamitans`, tally the numbner 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_ves, n) %>%
  filter(species_ves == "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
rv_pond	7	19	20	0	3	3
tuttle_pond	13	10	5	NA	NA	NA
vorisek_pond	2	9	4	NA	NA	NA
wood_lab_pond	4	NA	NA	NA	NA	NA

static n-mixture with no co-variates - 3 repeat visits

```
first_green_frog_unmarked_nmixture <- green_frog_visits %>%
  ungroup() %>%
  select(!c(site, c(visit_4:visit_6)))

second_green_frog_unmarked_nxmixture <- green_frog_visits %>%
 ungroup() %>%
  select(site, c(visit_4:visit_6)) %>%
  slice(1) %>%
  rename(visit_1 = visit_4,
         visit_2 = visit_5,
         visit_3 = visit_6) %>%
  select(!c(site))

green_frog_nmixture_3_visits <- rbind(first_green_frog_unmarked_nmixture, second_green_fro


umf_green3 <- unmarkedFramePCount(y = green_frog_nmixture_3_visits)

summary(umf_green3)
```

```
unmarkedFrame Object

5 sites
Maximum number of observations per site: 3
Mean number of observations per site: 2.6
Sites with at least one detection: 5

Tabulation of y observations:
   0    2    3    4    5    7    9   10   13   19   20 <NA>
   1    1    2    2    1    1    1    1    1    1    1    2
```

```
fm1_g3 <- pcount(~1 ~1, data = umf_green3)
```

```
Warning in pcount(~1 ~ 1, data = umf_green3): K was not specified and was set to
120.
```

```
backTransform(fm1_g3, "state")
```

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

 Estimate   SE LinComb (Intercept)
     24.4 9.67     3.2           1

Transformation: exp
```

```
backTransform(fm1_g3, "det")
```

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

 Estimate    SE LinComb (Intercept)
    0.305 0.118  -0.823           1

Transformation: logistic
```

Assuming the sites and enviro variables are exactly the same we estimate there is 24.42 +-9.67 frogs at each sites and with 31% of detecting each individual when 3 repeat visits occur nightly.

static n-mixture with no co-variates - 2 repeat visits

```
first_green_frog_unmarked_nmixture <- green_frog_visits %>%
  ungroup() %>%
  select(!c(site, c(visit_3:visit_6))) %>%
  slice(1:3)

second_green_frog_unmarked_nxmixture <- green_frog_visits %>%
 ungroup() %>%
  select(site, c(visit_4:visit_5)) %>%
  slice(1) %>%
  rename(visit_1 = visit_4,
         visit_2 = visit_5) %>%
  select(!c(site))

green_frog_nmixture_2_visits <- rbind(first_green_frog_unmarked_nmixture, second_green_fro


umf_green2 <- unmarkedFramePCount(y = green_frog_nmixture_2_visits)

summary(umf_green2)
```

```
unmarkedFrame Object

4 sites
Maximum number of observations per site: 2
Mean number of observations per site: 2
Sites with at least one detection: 4

Tabulation of y observations:
 0  2  3  7  9 10 13 19 
 1  1  1  1  1  1  1  1 
```

```
fm1_g2 <- pcount(~1 ~1, data = umf_green2)
```

```
Warning in pcount(~1 ~ 1, data = umf_green2): K was not specified and was set to
119.
```

```
backTransform(fm1_g2, "state")
```

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

 Estimate   SE LinComb (Intercept)
     45.4 50.3    3.81           1

Transformation: exp
```

```
backTransform(fm1_g2, "det")
```

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

 Estimate    SE LinComb (Intercept)
    0.174 0.193   -1.56           1

Transformation: logistic
```

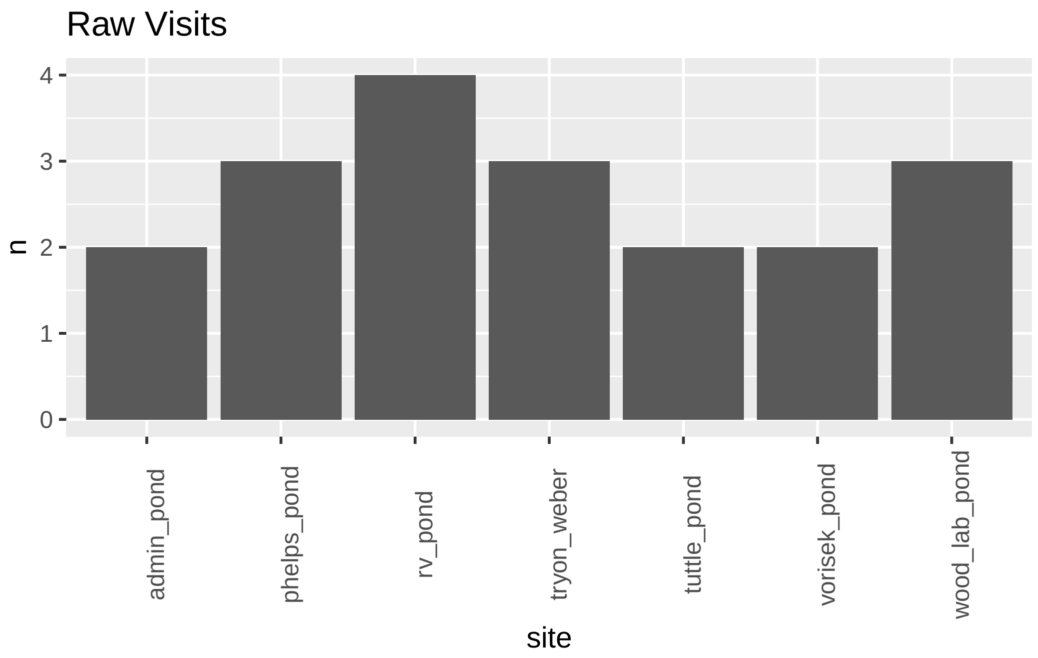
Assuming the sites and enviro variables are exactly the same we estimate there is 45.37 +-50.33 frogs at each sites and with 17% of detecting each individual when 3 repeat visits occur nightly.

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.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 > '2023-01-01';"

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, s2.detection_type
                from region r
                join site s on r.region_id = s.region_id
                join visit v on s.site_id = v.site_id
                join survey s2 on v.visit_id = s2.visit_id
                where r.region = 'pennsylvania'
                and v.date > '2023-01-01'
                and s2.detection_type = 'capture';"

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

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

Visualize all CMR Visits

```
viz <- cmr_raw_visits %>%
  group_by(site) %>%
  summarise(n = n())

ggplot(data = viz) +
  geom_col(aes(x=site, y = n)) +
  ggtitle("Raw Visits") +
  theme(axis.text.x = element_text(angle = 90))
```



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_type)) %>%
  unite(species_capture, c(species_capture, cmr_id), sep = "_")

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)) %>%
  drop_na(species_capture) %>%
  pivot_wider(names_from = c("n_visit"), values_from = c("n")) %>%
  #select(!c(visit_5, visit_6)) %>%
  # relocate(visit_1, .before = visit_2) %>%
  # relocate(visit_3, .after = visit_2) %>%

  # mutate(across(contains("visit"),
  #               ~ifelse(is.na(.) &
  #                          !is.na(lag(.)), 0, .))) %>%
  #select(sort(names(.))) %>%
  #filter(!site == "tryon_weber") %>%
  unite(species_capture, c("site", "species_capture"), sep = "_", na.rm = T) %>%
  mutate_all(~replace_na(.,0))
```

Final CMR matrix - 3 repeat Visits

```
clean_bull_3 <- clean_bull

clean_bull_3$captureHistory <- paste(clean_bull_3$visit_1, clean_bull_3$visit_2, clean_bul
                               sep = "")
```

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

lev <- unique(clean_bull_3$captureHistory)

clean_bull_3$captureHistory <- factor(clean_bull_3$captureHistory, levels = lev)

bull_table_3 <- table(clean_bull_3$species_capture, clean_bull_3$captureHistory)


kable(head(bull_table_3, n = 15))
```

	100	010	001	111	110	011
admin_pond_rana_catesbeiana_A2	1	0	0	0	0	0
admin_pond_rana_catesbeiana_A3	1	0	0	0	0	0
admin_pond_rana_catesbeiana_A4	0	1	0	0	0	0
phelps_pond_rana_catesbeiana_A2A3	0	0	1	0	0	0
rv_pond_rana_catesbeiana_A2	1	0	0	0	0	0
rv_pond_rana_catesbeiana_A3	0	0	0	1	0	0
rv_pond_rana_catesbeiana_A4	0	1	0	0	0	0
rv_pond_rana_catesbeiana_B3A2A3	0	0	1	0	0	0
tuttle_pond_rana_catesbeiana_A2	0	0	0	0	1	0
tuttle_pond_rana_catesbeiana_A2A3	1	0	0	0	0	0
tuttle_pond_rana_catesbeiana_A2A4	1	0	0	0	0	0
tuttle_pond_rana_catesbeiana_A2B2	1	0	0	0	0	0
tuttle_pond_rana_catesbeiana_A2B3	1	0	0	0	0	0
tuttle_pond_rana_catesbeiana_A2B4	1	0	0	0	0	0
tuttle_pond_rana_catesbeiana_A3	0	0	0	0	1	0

obs covariates, piFun equal detection, Model runn

```r
class(bull_table_3) <- "matrix"

o2y <- matrix(1, 3, 6)

crPiFun <- function(p) {
    p1 <- p[,1]
    p2 <- p[,2]
    p3 <- p[,3]

    cbind("100" = p1 * (1-p2) * (1-p3),    #1: 100
          "010" = p1 * (1-p2) * (1-p3),    #2: 010
          "001" = (1-p1) * (1-p2) * p3,    #3: 001
          "111" = p1 * p2 * p3,            #4: 111
          "110" = p1 * p2 * (1-p3),        #5: 110
          "011" = (1-p1) * p2 * p3         #6: 011

    )
}


umf_3 <- unmarkedFrameMPois(y = bull_table_3, piFun = "crPiFun", obsToY = o2y)

fm_3 <- multinomPois(~1 ~1, umf_3, engine = "R")

backTransform(fm_3, "state")
```

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

 Estimate   SE LinComb (Intercept)
     1.81 0.41   0.591           1

Transformation: exp
```

```
backTransform(fm_3, "det")
```

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

 Estimate     SE LinComb (Intercept)
    0.267 0.0649   -1.01           1

Transformation: logistic
```

Assuming sites are identical we estimate there are 1.86 individuals at each site? With a 26% chance of detecting each individual? Not fully sure how to interpret results.

Final CMR matrix - 2 repeat Visits

```
clean_bull_2 <- clean_bull

clean_bull_2$captureHistory <- paste(clean_bull_2$visit_1, clean_bull_2$visit_2,
                                     sep = "")

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

lev <- unique(clean_bull_2$captureHistory)

clean_bull_2$captureHistory <- factor(clean_bull_2$captureHistory, levels = lev)

bull_table_2 <- table(clean_bull_2$species_capture, clean_bull_2$captureHistory)


kable(head(bull_table_2, n = 15))
```

	10	01	00	11
admin_pond_rana_catesbeiana_A2	1	0	0	0
admin_pond_rana_catesbeiana_A3	1	0	0	0
admin_pond_rana_catesbeiana_A4	0	1	0	0
phelps_pond_rana_catesbeiana_A2A3	0	0	1	0
rv_pond_rana_catesbeiana_A2	1	0	0	0
rv_pond_rana_catesbeiana_A3	0	0	0	1
rv_pond_rana_catesbeiana_A4	0	1	0	0
rv_pond_rana_catesbeiana_B3A2A3	0	0	1	0
tuttle_pond_rana_catesbeiana_A2	0	0	0	1
tuttle_pond_rana_catesbeiana_A2A3	1	0	0	0
tuttle_pond_rana_catesbeiana_A2A4	1	0	0	0
tuttle_pond_rana_catesbeiana_A2B2	1	0	0	0
tuttle_pond_rana_catesbeiana_A2B3	1	0	0	0
tuttle_pond_rana_catesbeiana_A2B4	1	0	0	0
tuttle_pond_rana_catesbeiana_A3	0	0	0	1

obs covariates, piFun equal detection, Model runn

```r
class(bull_table_2) <- "matrix"

o2y <- matrix(1, 2, 4)

crPiFun <- function(p) {
    p1 <- p[,1]
    p2 <- p[,2]

    cbind("10" = p1 * (1-p2),        #1: 10
          "01" = p1 * (1-p2),        #2: 01
          "00" = (1-p1) * (1-p2),    #3: 00
          "11" = p1 * p2             #4: 11

    )
}


umf_2 <- unmarkedFrameMPois(y = bull_table_2, piFun = "crPiFun", obsToY = o2y)

fm_2 <- multinomPois(~1 ~1, umf_2, engine = "R")

backTransform(fm_2, "state")
```

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

 Estimate    SE  LinComb (Intercept)
        1 0.169 4.09e-05           1

Transformation: exp
```

```r
backTransform(fm_2, "det")
```

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

 Estimate     SE LinComb (Intercept)
    0.571 0.0591   0.288           1

Transformation: logistic
```

STOP: Species Not detected in 2023 data for VES

pseudacris_crucifer

N-Mixture Table formatting

Filter data for `pseudacris_crucifer`, tally the numbner 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_ves, n) %>%
#   filter(species_ves == "pseudacris_crucifer") %>%
#   select(!species_ves) %>%
#   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)
```

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

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

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_pipens

N-Mixture Table Formatting

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

```
# leop_frog_visits <- nmix_clean_up %>%
#   select(site, date, species_ves, n) %>%
#   filter(species_ves == "rana_pipiens") %>%
#   #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, .)))
#
#
# kable(leop_frog_visits)
```

static n-mixture with no co-variates

```
# leop_frog_unmarked_nmixture <- leop_frog_visits %>%
#   ungroup() %>%
#   select(!c(site))
#
#
# umf <- unmarkedFramePCount(y = leop_frog_unmarked_nmixture)
#
# summary(umf)
#
```

```
# fm1 <- pcount(~1 ~1, data = umf)
#
# backTransform(fm1, "state")
#
# backTransform(fm1, "det")
```