Patagonia parrots density analysis

Francisco Dénes and Peter Sólymos

June 2, 2018

Motivation

We want to model the density of the austral parakeet *Enicognathus ferrugineus* and the slender-billed parakeet *Enicognathus leptorhynchus* from count data obtained with road transect surveys in Patagonia. Because these parrots are gregarious, we want to assess whether group size and size of groups vary across habitats and between breeding (Nov-Dec) and non-breeding seasons (Jan-Oct). We use distance sampling methods to account for detectability of parrots.

These animals tend to form smaller or larger groups, but groups of size 2 are also often observed as a result of mating pairs, as shown in Figure 1.

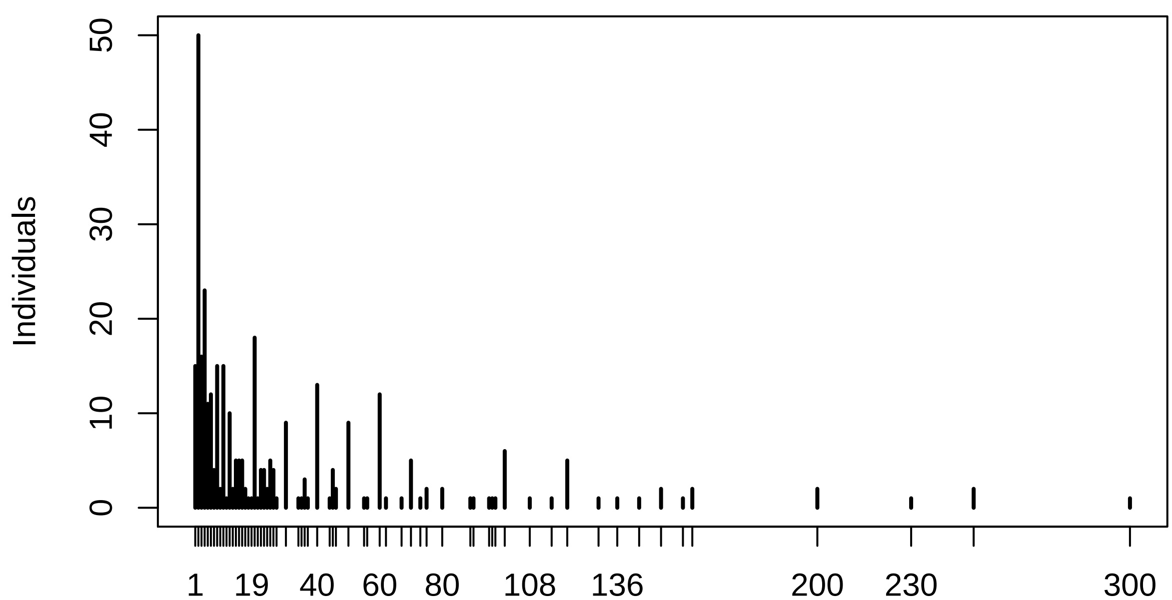

Figure 1: *Enicognathus ferrugineus* count frequencies

Austral parakeet *Enicognathus ferrugineus*

Estimating effective detection radius (EDR)

We fit distance sampling models with detection predictor variables, including the average group size, the number of groups that were observed (to account for the possibility that grouping behaviour influences detectability), and also habitat type. We used forward-stepwise model selection, starting with single covariate models and eliminating covariates that do not improve model parsimony (i.e. result in AIC lower than 2) in relation to the null model.

Table 1: *E. ferrugineous* EDR models AIC

	df	AIC	dAIC
EDR.habitatype	3	1434.532	0.00
EDR.null	1	1438.166	3.63
EDR.avggroupsize	2	1439.144	4.61
EDR.numbergroups	2	1440.166	5.63

The model (`EDR.habitatype`) has the lowest AIC (Table 1), indicating that habitat type affects the effective detection radius (EDR) of *Enicognathus ferrugineus*. Specifically, detection radius is higher in agropastoral than urban and other habitats (Table 2). This means that detectability is higher in agropastoral habitats, and also that the area surveyed in a given sample unit is larger if the habitat therein is agropastoral vs. urban or other, presubably because it is possible to see further in pastures and planted fields than in forest or urban environments.

```
##
## Call:
## cmulti(formula = Y | D ~ Urban + Agropastoral, data = X, type = "dis")
##
## Distance Sampling (half-normal, circular area)
## Conditional Maximum Likelihood estimates
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## log.tau_(Intercept)    4.58872    0.03844 119.365   <2e-16 ***
## log.tau_Urban          0.05744    0.05814   0.988   0.3232
## log.tau_Agropastoral  0.26269    0.10278   2.556   0.0106 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-likelihood: -714.3
## BIC =  1444
```

Table 2: *E. ferrugineous* habitat-specific EDR (m)

Habitat	EDR
Other	98.36847
Urban	104.18432
Agropastoral	127.92056

Models for number of groups

We then model the number of groups as a function of covariates. The model for number of groups is $G_i \sim \text{Poisson}(D_i A_i)$, where D_i = covariates and A_i = area sampled in site. A_i is calculated using the habitat-specific estimated EDR, and is added to the model as an offset. We build a first set of models to evaluate the effect of covariates related to habitat (habitat type and elevation).

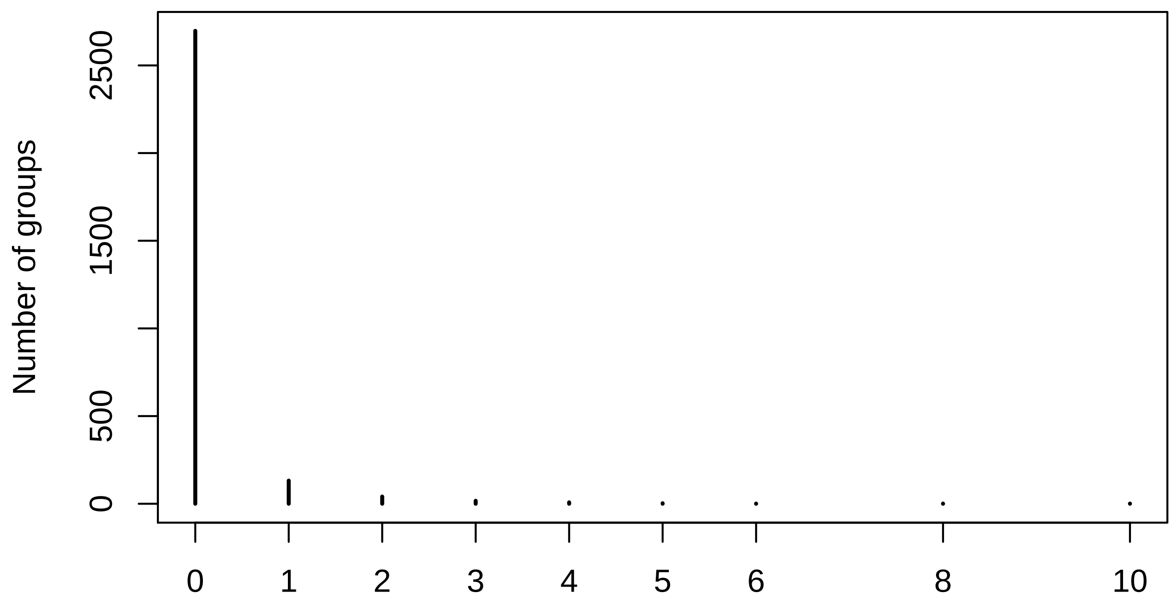

Figure 2: *Enicognathus ferrugineus* group numbers

Table 3: *E. ferrugineous* number of group models

	df	AIC	dAIC
ngroup.hab.ele2	5	2082.963	0.00
ngroup.hab.ele	4	2114.437	31.47
ngroup.hab	3	2133.717	50.75
ngroup.ele2	3	2275.565	192.60
ngroup.ele	2	2323.806	240.84

The model with both habitat type and elevation (quadratic effect) has the lowest AIC (Table 3), indicating that both covariates affect the number of groups:

```
##
## Call:
## glm(formula = ngroups ~ habitat + elevation, family = poisson,
##     data = sites, offset = log(sites$A))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9552  -0.3696  -0.2192  -0.1266   7.6615
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.8742985  0.2033144 -19.056  < 2e-16 ***
## habitatOther  1.7435771  0.1978580   8.812  < 2e-16 ***
## habitatUrban  2.4225361  0.2020368  11.991  < 2e-16 ***
## elevation    -0.0007506  0.0001637  -4.586 4.51e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1865.3  on 2900  degrees of freedom
## Residual deviance: 1636.1  on 2897  degrees of freedom
## AIC: 2114.4
##
## Number of Fisher Scoring iterations: 7
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: ngroups
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev
## NULL                       2900     1865.3
## habitat    2  207.965      2898     1657.4
## elevation  1   21.279      2897     1636.1
```

We then proceed by adding within-year temporal covariates (breeding/non-breeding season and julian date) and their interactions with habitat.

Table 4: *E. ferrugineous* number of group models (within-year temporal predictors) AIC

	df	AIC	dAIC
ngroup.habXseason.ele	8	2065.594	0.00
ngroup.hab.ele.season	6	2067.555	1.96
ngroup.habXjdate.ele	8	2081.546	15.95
ngroup.hab.ele.jdate	6	2084.491	18.90
ngroup.hab.ele	4	2114.437	48.84

Model with season*habitat interaction is equally parcimonious with the model with only the additive effects.

Given these are nested models, this indicates the interaction only marginally improves the model fit, so will continue with model with season and no interaction:

season*habitat interaction model

```
##
## Call:
## glm(formula = ngroups ~ elevation + I(elevation^2) + habitat *
##     season, family = poisson, data = sites, offset = log(sites$A))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1791  -0.3703  -0.2113  -0.1192   7.9062
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      -1.819e+01  5.580e+02  -0.033    0.974
## elevation                        -3.900e-03  5.957e-04  -6.547 5.88e-11 ***
## I(elevation^2)                    2.637e-06  4.795e-07   5.500 3.80e-08 ***
## habitatOther                      1.580e+01  5.580e+02   0.028    0.977
## habitatUrban                      2.008e+00  9.084e+02   0.002    0.998
## seasonnon-breeding                1.519e+01  5.580e+02   0.027    0.978
## habitatOther:seasonnon-breeding -1.436e+01  5.580e+02  -0.026    0.979
## habitatUrban:seasonnon-breeding  2.663e-01  9.084e+02   0.000    1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1865.3  on 2900  degrees of freedom
## Residual deviance: 1579.2  on 2893  degrees of freedom
## AIC: 2065.6
##
## Number of Fisher Scoring iterations: 16
```

no interaction between season and habitat

```
##
## Call:
## glm(formula = ngroups ~ habitat + elevation + I(elevation^2) +
##     season, family = poisson, data = sites, offset = log(sites$A))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1472  -0.3692  -0.2153  -0.1244   7.8798
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -4.250e+00  4.119e-01 -10.318  < 2e-16 ***
## habitatOther         1.510e+00  2.037e-01   7.411 1.25e-13 ***
## habitatUrban         2.314e+00  2.026e-01  11.424  < 2e-16 ***
## elevation           -3.800e-03  5.929e-04  -6.408 1.47e-10 ***
## I(elevation^2)       2.565e-06  4.781e-07   5.365 8.11e-08 ***
```

```
## seasonnon-breeding  1.175e+00  3.404e-01   3.451 0.000558 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1865.3  on 2900  degrees of freedom
## Residual deviance: 1585.2  on 2895  degrees of freedom
## AIC: 2067.6
##
## Number of Fisher Scoring iterations: 7
```

Finally, we assess year effects by adding a year covariate (2013-2016).

Table 5: *E. ferrugineous* number of group models (year predictor) AIC table

	df	AIC	dAIC
ngroup.hab.ele.season.year	9	2022.265	0.00
ngroup.hab.ele.season	6	2067.555	45.29

The model with lowest AIC indicates that the number of groups is affected by habitat ype, elevation, season (breeding/non breeding) and year.

```
##
## Call:
## glm(formula = ngroups ~ habitat + elevation + I(elevation^2) +
##     season + as.factor(year), family = poisson, data = sites,
##     offset = log(sites$A))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6688  -0.4066  -0.2324  -0.1305   7.4851
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -4.716e+00  4.170e-01 -11.311  < 2e-16 ***
## habitatOther         1.739e+00  2.062e-01   8.434  < 2e-16 ***
## habitatUrban         2.460e+00  2.037e-01  12.075  < 2e-16 ***
## elevation           -3.430e-03  6.005e-04  -5.712 1.12e-08 ***
## I(elevation^2)       2.230e-06  4.860e-07   4.588 4.47e-06 ***
## seasonnon-breeding   1.022e+00  3.479e-01   2.939  0.00329 **
## as.factor(year)2014  9.236e-02  1.683e-01   0.549  0.58307
## as.factor(year)2015  9.180e-01  1.614e-01   5.688 1.28e-08 ***
## as.factor(year)2016  6.115e-01  2.053e-01   2.979  0.00289 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1865.3  on 2900  degrees of freedom
## Residual deviance: 1533.9  on 2892  degrees of freedom
## AIC: 2022.3
##
```

```
## Number of Fisher Scoring iterations: 6
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: ngroups
##
## Terms added sequentially (first to last)
##
##
##                 Df Deviance Resid. Df Resid. Dev
## NULL                             2900     1865.3
## habitat          2  207.965      2898     1657.4
## elevation        1   21.279      2897     1636.1
## I(elevation^2)   1   33.475      2896     1602.6
## season           1   17.408      2895     1585.2
## as.factor(year)  3   51.290      2892     1533.9
```

Models for group size

The count distribution is characterized with a spike at 2 (Figure 1), and by the absence of 0s due to group size being conditional on having >0 birds to consider it a group.

We start developing a general V-Inflated Poisson (VIP) model ("V" stands for variable, in refence to the "Z" representing 0 in a zero-inflated model, or ZIP), then we add the >0 condition.

Maximum likelihood

Let Y be a random variable, and y are observations, V is the count value that has some extra probability mass ($V = 0$ is the ZIP model), $f(y; \lambda)$ is the Poisson density ($f(y; \lambda) = e^{-\lambda} \frac{\lambda^y}{y!}$).

The V-Inflated density can be written as $P(Y = y) = \phi I(Y = V) + (1 - \phi) f(y; \lambda)$ which is $\phi + (1 - \phi) f(V; \lambda)$ when $Y = V$ and $(1 - \phi) f(y; \lambda)$ otherwise.

R functions for the VIP model are presented at the end of this document. Simulations are done to check the estimating procedure.

We define the extra probability mass at `V=2` to account for the group size peak in pairs. The model is also truncated at 0, because there are no groups with 0 individuals. We include habitat type as a covariate for the count (Poisson) component, and compare models with and without season (breeding/non-breeding) as a covariate for the V-inflation probability.

Table 6: *E. ferrugineous* Group size models AIC table

	df	AIC	dAIC
VIP.habitat.season	331	32075.50	0.00
VIP.habitat	332	32085.51	10.01

```
##
## Call:
## vip(Y = x$count, X = X, Z = Z, V = 2, offsetx = log(x$A), truncate = TRUE,
##     hessian = TRUE, method = "SANN")
```

```
##
## V-Inflated (Zero-Truncated) Poisson Model
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## P_(Intercept)         1.00002    0.01898  52.686  < 2e-16 ***
## P_Urban               1.75433    0.02294  76.483  < 2e-16 ***
## P_Agropastoral        0.61163    0.03653  16.744  < 2e-16 ***
## V_(Intercept)         1.31556    0.81619   1.612    0.107
## V_seasonnon-breeding -3.47379    0.83871  -4.142 3.45e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-likelihood: -1.571e+04
## BIC = 3.334e+04
```

The model with season is more parsimonious (Table 6). Estimated model coefficients indicate that group sizes are larger in urban (β = 1.75) and agropastoral (β = 0.61) habitats when compared to other natural habitats. Moreover, the probability of there being extra pairs (i.e. the V-inflation probability, ϕ) is smaller in the non-breeding season (β = -3.47).

Slender-billed parakeet *Enicognathus leptorhynchus*

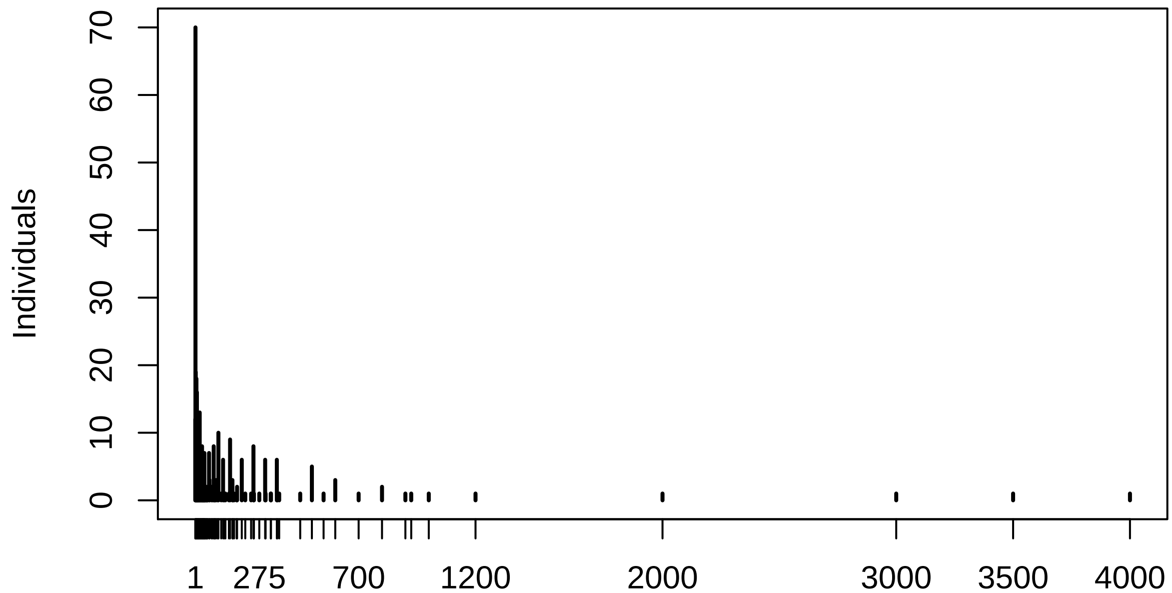

Figure 3: *Enicognathus leptorhynchus* count frequencies

Estimating effective detection radius (EDR)

Table 7: *E. ferrugineous* EDR models AIC

	df	AIC	dAIC
EDR.avggroupsize.habitat	4	1054.464	0.00
EDR.habitat.avggroupsize.numbergroups	5	1059.062	4.60
EDR.avggroupsize	2	1063.875	9.41
EDR.avggroupsize.numbergroups	3	1064.553	10.09
EDR.habitat.numbergroups	4	1069.376	14.91
EDR.habitatype	3	1070.755	16.29
EDR.null	1	1086.330	31.87
EDR.numbergroups	2	1087.172	32.71

The model (`EDR.avggroupsize.habitat`) has the lowest AIC (Table 7), indicating that habitat type and average group size affect the effective detection radius (EDR) of *Enicognathus leptorhynchus*. The mean EDR for each habitat, predicted using model coefficients and the habitat-specific means of average group size, is shown in Table 8.

```
##
## Call:
## cmulti(formula = Y | D ~ gavg + Urban + Agropastoral, data = X,
##     type = "dis")
##
## Distance Sampling (half-normal, circular area)
## Conditional Maximum Likelihood estimates
##
## Coefficients:
##                        Estimate Std. Error    z value Pr(>|z|)
## log.tau_(Intercept)   5.361e+00  4.209e-51  1.274e+51   <2e-16 ***
## log.tau_gavg          8.091e-04  8.419e-55  9.611e+50   <2e-16 ***
## log.tau_Urban        -2.655e-01  4.209e-51 -6.308e+49   <2e-16 ***
## log.tau_Agropastoral -2.883e-02  5.953e-51 -4.843e+48   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-likelihood: -523.2
## BIC =  1067
```

Table 8: *E. leptorhynchus* habitat-specific mean EDR (m)

Habitat	EDR
Other	235.4183
Urban	224.2124
Agropastoral	180.7002

Models for number of groups

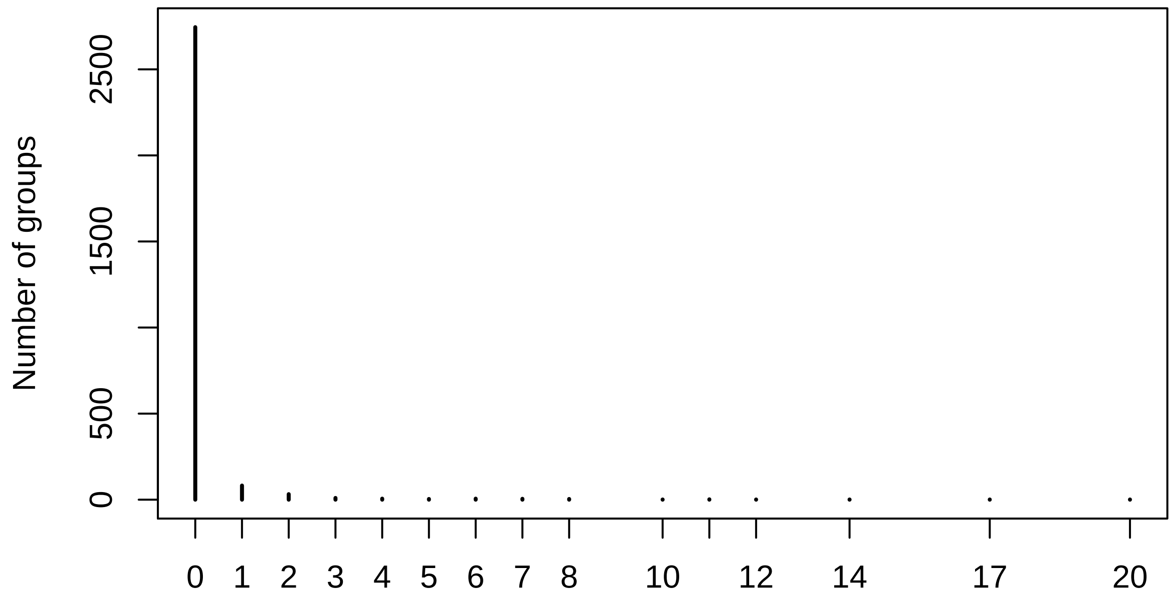

Figure 4: *Enicognathus leptorhynchus* group numbers

First set of models to evaluate the effect of covariates related to habitat (habitat type and elevation):

Table 9: *E. leptorhynchus* number of group models

	df	AIC	dAIC
ngroup.hab.ele2	5	2486.990	-0.84
ngroup.hab.ele	4	2487.834	0.00
ngroup.hab	3	2493.346	5.51
ngroup.ele2	3	2746.031	258.20
ngroup.ele	2	2761.122	273.29

The models with and habitat type and elevation (linear effect)habitat type and elevation (quadratic effect) are equally parsimonious (Table 9). Given their nestedness, we drop the quadratic elevation effect and continue with the model with habitat and linear elevation effects.

```
##
## Call:
## glm(formula = ngroups ~ habitat + elevation, family = poisson,
```

```
##     data = sites, offset = log(sites$A))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.7079  -0.4007  -0.2157  -0.1141  10.6088
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.7809811  0.1232247 -22.568  < 2e-16 ***
## habitatOther -2.0346998  0.1466592 -13.874  < 2e-16 ***
## habitatUrban -0.7341155  0.1434994  -5.116 3.12e-07 ***
## elevation     0.0004781  0.0001748   2.736  0.00622 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2358.9  on 2900  degrees of freedom
## Residual deviance: 2081.0  on 2897  degrees of freedom
## AIC: 2487.8
##
## Number of Fisher Scoring iterations: 7
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: ngroups
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev
## NULL                       2900     2358.9
## habitat    2  270.326      2898     2088.5
## elevation  1    7.512      2897     2081.0
```

Adding within-year temporal covariates (breeding/non-breeding season and julian date) and their interactions with habitat:

Table 10: *E. leptorhynchus* number of group models (within-year temporal predictors) AIC

	df	AIC	dAIC
ngroup.habXjdate.ele	7	2110.653	0.00
ngroup.hab.ele.jdate	5	2230.345	119.69
ngroup.habXseason.ele	7	2472.731	362.08
ngroup.hab.ele	4	2487.834	377.18
ngroup.hab.ele.season	5	2489.059	378.41

The model with jdate*habitat interaction has the lowest AIC (Table 9).

season*habitat interaction model

```
##
## Call:
## glm(formula = ngroups ~ elevation + habitat * jdate, family = poisson,
##     data = sites, offset = log(sites$A))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.6867  -0.3765  -0.0688  -0.0037   9.0194
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.4633740  0.1674177  -8.741  < 2e-16 ***
## elevation            0.0003340  0.0001723   1.939 0.052549 .
## habitatOther         2.2340994  0.5897471   3.788 0.000152 ***
## habitatUrban         0.4120594  0.3099620   1.329 0.183720
## jdate               -0.0057427  0.0005726 -10.029  < 2e-16 ***
## habitatOther:jdate -0.0399419  0.0071459  -5.589 2.28e-08 ***
## habitatUrban:jdate -0.0093727  0.0022889  -4.095 4.22e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2358.9  on 2900  degrees of freedom
## Residual deviance: 1697.8  on 2894  degrees of freedom
## AIC: 2110.7
##
## Number of Fisher Scoring iterations: 10
```

Assess year effects by adding a year covariate (2013-2016).

Table 11: *E. leptorhynchus* number of group models (year predictor) AIC table

	df	AIC	dAIC
ngroup.habXjdate.ele.year	8	1993.981	0.00
ngroup.habXjdate.ele	7	2110.653	116.67

The model with lowest AIC indicates that the number of groups is affected by habitat type, elevation, julian date and year.

```
##
## Call:
## glm(formula = ngroups ~ habitat + elevation + season + as.factor(year),
##     family = poisson, data = sites, offset = log(sites$A))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9000  -0.4038  -0.2032  -0.1057   8.6857
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)         -3.9524864  0.2639635 -14.974  < 2e-16 ***
## habitatOther        -1.6293917  0.1495803 -10.893  < 2e-16 ***
## habitatUrban        -0.5062433  0.1443452  -3.507 0.000453 ***
## elevation            0.0004645  0.0001629   2.851 0.004355 **
## seasonnon-breeding  -0.7487438  0.2010999  -3.723 0.000197 ***
## as.factor(year)2014  0.6694517  0.2764278   2.422 0.015444 *
## as.factor(year)2015  1.7634765  0.2366430   7.452 9.19e-14 ***
## as.factor(year)2016  3.2799760  0.2393254  13.705  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2358.9  on 2900  degrees of freedom
## Residual deviance: 1579.2  on 2893  degrees of freedom
## AIC: 1994
##
## Number of Fisher Scoring iterations: 7
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: ngroups
##
## Terms added sequentially (first to last)
##
##
##                 Df Deviance Resid. Df Resid. Dev
## NULL                             2900     2358.9
## habitat          2   270.33      2898     2088.5
## elevation        1     7.51      2897     2081.0
## season           1     0.78      2896     2080.2
## as.factor(year)  3   501.08      2893     1579.2
```

Models for group size

```
##      P_(Intercept)     P_Urban P_Agropastoral V_(Intercept)
## [1,]    -3.4315607 24.02749982      17.308699     5.5023551
## [2,]    -1.6055318  5.64347167     -17.089005    13.1416174
## [3,]   -23.5915496 23.21956971      -6.478753    10.3184573
## [4,]     0.9674519  5.99061566       6.542012    16.3476814
## [5,]   -32.3644298 -0.01516136     -13.319216     0.8403038
## [6,]   -15.0687507 18.07756751       4.587351    49.1140564
##      V_seasonnon-breeding
## [1,]             11.61975
## [2,]             25.88478
## [3,]             30.86020
## [4,]             15.35033
## [5,]             43.74769
## [6,]             26.71279
```

		5%	95%
P_(Intercept)	15.5148542	-34.103307	7.173057
P_Urban	-8.6162679	-11.783613	28.778820
P_Agropastoral	-0.0262193	-20.672452	19.980277
V_(Intercept)	-23.5420846	4.795528	45.179416
V_seasonnon-breeding	-22.8068590	1.472501	46.768732

VIP model - R functions and simulations

```
vip <-
function(Y, X, Z, V=0,
offsetx, offsetz, weights, linkz="logit",
truncate=FALSE, hessian=TRUE, method="Nelder-Mead", init=NULL, ...) {
    if (missing(Y))
        stop("C'mon, you must have some data?!")
    if (truncate && any(Y < 1))
        stop("Y must be >0 when truncate=TRUE")
    n <- length(Y)
    id0 <- Y == V
    id1 <- !id0
    if (missing(X)) {
        X <- matrix(1, n, 1)
        colnames(X) <- "(Intercept)"
    }
    if (missing(Z)) {
        Z <- matrix(1, n, 1)
        colnames(Z) <- "(Intercept)"
    }
    kx <- ncol(X)
    kz <- ncol(Z)
    if (missing(offsetx))
        offsetx <- 0
    if (missing(offsetz))
        offsetz <- 0
    if (missing(weights))
        weights <- rep(1, n)
    linkinvx <- poisson("log")$linkinv
    linkinvz <- binomial(linkz)$linkinv
    good.num.limit <- c(.Machine$double.xmin, .Machine$double.xmax)^(1/3)

    ## VIP model full likelihood
    nll_VIP_ML <- function(parms) {
        mu <- as.vector(linkinvx(X %*% parms[1:kx] + offsetx))
        phi <- as.vector(linkinvz(Z %*% parms[(kx + 1):(kx + kz)] + offsetz))
        loglik0 <- log(phi + (1 - phi) * dpois(V, lambda = mu, log = FALSE))
        loglik1 <- log(1 - phi) + dpois(Y, lambda = mu, log = TRUE)
        loglik <- sum(weights[id0] * loglik0[id0]) + sum(weights[id1] * loglik1[id1])
        if (!is.finite(loglik) || is.na(loglik))
            loglik <- -good.num.limit[2]
```

```
        -loglik
    }
    ## 0-truncated VIP model full likelihood
    nll_VIP_TR <- function(parms) {
        mu <- as.vector(linkinvx(X %*% parms[1:kx] + offsetx))
        phi <- as.vector(linkinvz(Z %*% parms[(kx + 1):(kx + kz)] + offsetz))
        loglik0 <- log(phi + (1 - phi) * dpois(V, lambda = mu, log = FALSE) / (1-exp(-mu)))
        loglik1 <- log((1 - phi) * dpois(Y, lambda = mu, log = FALSE) / (1-exp(-mu)))
        loglik <- sum(weights[id0] * loglik0[id0]) + sum(weights[id1] * loglik1[id1])
        if (!is.finite(loglik) || is.na(loglik))
            loglik <- -good.num.limit[2]
        -loglik
    }

    if (is.null(init))
      init <- rep(0, kx+kz)
    opt <- optim(init,
        if (truncate) nll_VIP_TR else nll_VIP_ML,
        hessian=hessian, method=method, ...)
    par <- opt$par
    names(par) <- c(paste0("P_", colnames(X)), paste0("V_", colnames(Z)))
    vc <- if (hessian)
        solve(opt$hessian) else matrix(NA, length(par), length(par))
    dimnames(vc) <- list(names(par), names(par))
    out <- list(call=match.call(),
        coefficients=par, loglik=-opt$value, vcov=vc, nobs=n,
        truncate=truncate)
    class(out) <- "vip"
    out
}
vcov.vip <- function(object, ...) object$vcov
logLik.vip <- function (object, ...)
    structure(object$loglik, df = object$nobs - length(object$coef),
        nobs = object$nobs, class = "logLik")
summary.vip <- function (object, ...) {
    k <- length(object$coefficients)
    coefs <- coef(object)
    se <- sqrt(diag(vcov(object)))
    tstat <- coefs/se
    pval <- 2 * pnorm(-abs(tstat))
    coefs <- cbind(coefs, se, tstat, pval)
    colnames(coefs) <- c("Estimate", "Std. Error", "z value", "Pr(>|z|)")
    coefs <- coefs[1:k, , drop = FALSE]
    rownames(coefs) <- names(coef(object))
    out <- list(call = object$call, coefficients=coefs, loglik = object$loglik,
        bic=BIC(object), truncate=object$truncate)
    class(out) <- "summary.vip"
    return(out)
}
print.summary.vip <- function (x, digits, ...)
{
    if (missing(digits))
        digits <- max(3, getOption("digits") - 3)
```

```r
    cat("\nCall:", deparse(x$call,
        width.cutoff = floor(getOption("width") * 0.85)), "", sep = "\n")
    cat("V-Inflated", if (x$truncate) "(Zero-Truncated)" else "", "Poisson Model\n\n")
    cat(paste("Coefficients:\n", sep = ""))
    printCoefmat(x$coefficients, digits = digits, signif.legend = FALSE)
    if (!any(is.na(array(x$coefficients)))) {
        if (getOption("show.signif.stars") & any(x$coefficients[,4] < 0.1))
            cat("---\nSignif. codes: ", "0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1", "\n")
    }
    cat("\nLog-likelihood:", formatC(x$loglik, digits = digits),
        "\nBIC =", formatC(x$bic, digits = digits), "\n")
    cat("\n")
    invisible(x)
}
confint.vip <-
function (object, parm, level = 0.95, ...)
{
    cf <- coef(object)
    pnames <- names(cf)
    if (missing(parm)) {
        parm <- pnames
    } else {
        if (is.numeric(parm))
            parm <- pnames[parm]
    }
    a <- (1 - level)/2
    a <- c(a, 1 - a)
    pct <- paste(format(100 * a, trim = TRUE, scientific = FALSE, digits = 3), "%", sep="")
    ci <- array(NA, dim = c(length(parm), 2), dimnames = list(parm, pct))
    fac <- qnorm(a)
    ses <- sqrt(diag(vcov(object, model, type)))
    ci[] <- cf[parm] + ses[parm] %o% fac
    ci
}
```

Simple case

```r
set.seed(123)
n <- 1000
lam <- 2 # poisson mean, can be a vector of length n
phi <- 0.4 # V-inflation probability, can be a vector of length n
V <- 2 # V is the count value, can be 0, 2, etc
y <- y0 <- rpois(n, lam)
a <- rbinom(n, 1, phi)
y[a > 0] <- V
table(Poisson=y0, Vinflated=y)
```

```
##        Vinflated
## Poisson   0   1   2   3   4   5   6   8
##       0  81   0  51   0   0   0   0   0
##       1   0 151 126   0   0   0   0   0
##       2   0   0 274   0   0   0   0   0
```

```
##        3    0    0   65  112    0    0    0    0
##        4    0    0   39    0   43    0    0    0
##        5    0    0   12    0    0   29    0    0
##        6    0    0    6    0    0    0    9    0
##        7    0    0    1    0    0    0    0    0
##        8    0    0    0    0    0    0    0    1
```

```
mod <- vip(Y=y, V=2)
summary(mod)
```

```
##
## Call:
## vip(Y = y, V = 2)
##
## V-Inflated  Poisson Model
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## P_(Intercept)  0.70472    0.02909  24.224  < 2e-16 ***
## V_(Intercept) -0.33900    0.08824  -3.842 0.000122 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-likelihood: -1345
## BIC =  9585
```

```
cbind(True=c(log_lam=log(lam), logit_phi=qlogis(phi)),
      Est=coef(mod))
```

```
##                 True        Est
## log_lam    0.6931472  0.7047243
## logit_phi -0.4054651 -0.3389963
```

Covariates for the non-V part

```
set.seed(123)
n <- 10000
x <- rnorm(n)
df <- data.frame(x=x)
X <- model.matrix(~x, df)
beta <- c(-0.5,-0.5) # Intercept and beta values for covariate
lam <- exp(X %*% beta) # poisson mean, can be a vector of length n
phi <- 0.4 # V-inflation probability, can be a vector of length n
V <- 2 # V is the count value, can be 0, 2, etc
y <- y0 <- rpois(n, lam)
a <- rbinom(n, 1, phi)
y[a > 0] <- V
table(Poisson=y0, Vinflated=y)
```

```
##          Vinflated
## Poisson      0    1    2    3    4    5    6    7    8
##        0  3182    0 2131    0    0    0    0    0    0
##        1     0 1981 1137    0    0    0    0    0    0
##        2     0    0 1088    0    0    0    0    0    0
```

```
##       3    0    0  118  226    0    0    0    0    0
##       4    0    0   40    0   57    0    0    0    0
##       5    0    0   14    0    0   17    0    0    0
##       6    0    0    1    0    0    0    3    0    0
##       7    0    0    2    0    0    0    0    1    0
##       8    0    0    1    0    0    0    0    0    1
```

```r
mod <- vip(Y=y, X=X, V=2)
summary(mod)
```

```
##
## Call:
## vip(Y = y, X = X, V = 2)
##
## V-Inflated  Poisson Model
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## P_(Intercept) -0.45313    0.02037  -22.24   <2e-16 ***
## P_x           -0.49231    0.01664  -29.58   <2e-16 ***
## V_(Intercept) -0.48770    0.02483  -19.64   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-likelihood: -1.133e+04
## BIC = 1.147e+05
```

```r
cbind(True=c(beta=beta, logit_phi=qlogis(phi)),
      Est=coef(mod))
```

```
##                 True        Est
## beta1     -0.5000000 -0.4531273
## beta2     -0.5000000 -0.4923100
## logit_phi -0.4054651 -0.4876957
```

Methods

```r
coef(mod)
```

```
## P_(Intercept)           P_x V_(Intercept)
##    -0.4531273    -0.4923100    -0.4876957
```

```r
vcov(mod)
```

```
##               P_(Intercept)           P_x V_(Intercept)
## P_(Intercept)  0.0004151059  1.815322e-04 -1.454395e-04
## P_x            0.0001815322  2.769780e-04 -5.339019e-05
## V_(Intercept) -0.0001454395 -5.339019e-05  6.165031e-04
```

```r
summary(mod)
```

```
##
## Call:
## vip(Y = y, X = X, V = 2)
##
## V-Inflated  Poisson Model
```

```
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## P_(Intercept) -0.45313    0.02037  -22.24   <2e-16 ***
## P_x           -0.49231    0.01664  -29.58   <2e-16 ***
## V_(Intercept) -0.48770    0.02483  -19.64   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-likelihood: -1.133e+04
## BIC = 1.147e+05
```

```
confint(mod)
```

```
##                    2.5%      97.5%
## P_(Intercept) -0.4930599 -0.4131947
## P_x           -0.5249290 -0.4596910
## V_(Intercept) -0.5363606 -0.4390308
```

```
nobs(mod)
```

```
## [1] 10000
```

```
logLik(mod)
```

```
## 'log Lik.' -11332.89 (df=9997)
```

```
AIC(mod)
```

```
## [1] 42659.77
```

```
BIC(mod)
```

```
## [1] 114741.5
```

Zero-truncated VIP

We can truncate counts to be larger than 0. We also need $V > 0$ (for $V = 0$ case, look into ZIP or conditional Poisson model). Conceptually, the V-Inflation follows the 0-truncation (because we cannot observe 0, real truncated distribution).

The 0-truncated PDF is $P(Y = y \mid Y > 0) = \frac{P(Y=y)}{1-P(Y=0)}$. The 0-truncated V-Inflated density is $P(Y = y \mid Y > 0, V > 0) = \phi I(Y = V) + (1-\phi)\frac{f(y;\lambda)}{1-f(0;\lambda)}$. This can be achieved in the `vip` call by the argument `truncate=TRUE`.

Here we use covariates for both the V and non-V part.

```
set.seed(1)
n <- 1000
x <- rnorm(n)
z <- runif(n, -1, 1)
df <- data.frame(x=x, z=z)
X <- model.matrix(~x, df)
Z <- model.matrix(~z, df)
beta <- c(-0.5, -0.5)
alpha <- c(0, 0.5)
lam <- exp(X %*% beta)
```

```r
phi <- plogis(Z %*% alpha)
V <- 2 # V is the count value, cannot be 0
y <- y0 <- rpois(n, lam)
a <- rbinom(n, 1, phi)
keep <- y0>0
y <- y[keep] # conditioning (i.e. exclude 0s)
y0 <- y0[keep]
X <- X[keep,]
Z <- Z[keep,]
y[a[keep] > 0] <- V
table(Poisson=y0, Vinflated=y)
```

```
##        Vinflated
## Poisson   1   2   3   4   6
##       1 155 141   0   0   0
##       2   0 127   0   0   0
##       3   0  21  16   0   0
##       4   0   4   0   7   0
##       5   0   2   0   0   0
##       6   0   0   0   0   1
```

```r
mod <- vip(Y=y, X=X, Z=Z, V=2, truncate=TRUE)
summary(mod)
```

```
##
## Call:
## vip(Y = y, X = X, Z = Z, V = 2, truncate = TRUE)
##
## V-Inflated (Zero-Truncated) Poisson Model
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## P_(Intercept) -0.50814    0.14803  -3.433 0.000598 ***
## P_x           -0.57344    0.10170  -5.639 1.71e-08 ***
## V_(Intercept)  0.02131    0.12572   0.170 0.865387
## V_z            0.47041    0.20691   2.273 0.022999 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-likelihood: -384.1
## BIC =  3664
```

```r
cbind(True=c(beta=beta, alpha=alpha),
      Est=coef(mod))
```

```
##        True         Est
## beta1  -0.5 -0.50813933
## beta2  -0.5 -0.57343540
## alpha1  0.0  0.02131236
## alpha2  0.5  0.47040670
```