Overview of Unmarked:
An R Package for the Analysis of Wildlife Data

Ian Fiske and Richard Chandler

March 16, 2010

Abstract

Unmarked aims to be a complete environment for the statistical analysis of wildlife data. Currently, the focus is on 2-level hierarchical models that separately model a latent state and an observation process. Unmarked uses S4 classes to help the user both explore their data and analyze in a transparent manner.

1 Overview of unmarked

Occupancy and abundance data are often associated with metadata related to the design of the study. For example, in distance sampling, the study design (line- or point-transect), distance class break points, transect lengths, and units of measurement need to be accounted for in the analysis. Unmarked uses S4 classes to store data and metadata in a way that allows for easy data manipulation, summarization, and model specification. Table 1 lists the currently implemented models and their associated fitting functions and data classes.

Each data class can be created with a call to the constructor function of the same name as described in the examples below.

2 Typical unmarked session

The first step is to import the data into R. This can be accomplished with either a call to the appropriate type of unmarkedFrame:

```
> library(unmarked)
> wt <- read.csv(system.file("csv", "widewt.csv", package = "unmarked"))
> head(wt)
  site y.1 y.2 y.3       elev       forest   length    date.1
1    1   0   0   0 -1.1729446 -1.156228147 1.824549 -1.761481
2    2   0   0   0 -1.1265010 -0.501483710 1.629241 -2.904339
3    3   0   0   0 -0.1976283 -0.101362109 1.458615 -1.690053
4    4   0   0   0 -0.1047411  0.007761963 1.686399 -2.190053
5    5   0   0   0 -1.0336137 -1.192602838 1.280934 -1.832910
6    6   0   0   0 -0.8478392  0.917129237 1.808289 -2.618624
      date.2    date.3     ivel.1     ivel.2     ivel.3
1  0.3099471 1.3813757 -0.5060353 -0.5060353 -0.5060353
2 -1.0471958 0.5956614 -0.9336151 -0.9907486 -1.1621491
3 -0.4757672 1.4528042 -1.1355754 -1.3388644 -1.6099164
```

Model	Fitting Function	Data	Citation
Occupancy	occu	unmarkedFrameOccu	[2]
Royle-Nichols	occuRN	unmarkedFrameOccu	[5]
Point Count	pcount	unmarkedFramePCount	[4]
Distance-sampling	distsamp	unmarkedFrameDS	[6]
Arbitrary multinomial-Poisson	multinomPois	unmarkedFrameMPois	[3]
Colonization-extinction	colext	unmarkedMultFrame	[1]

Table 1: Models handled by unmarked.

```
4 -0.6900529 1.2385185 -0.8193481 -0.9272669 -1.1970640
5  0.1670899 1.3813757  0.6375563  0.8803737  1.0422520
6  0.1670899 1.3813757 -1.3288666 -1.0422624 -0.8989603
> y <- wt[, 2:4]
> siteCovs <- wt[, c("elev", "forest", "length")]
> obsCovs <- reshape(wt[, c("date.1", "date.2", "date.3",
     "ivel.1", "ivel.2", "ivel.3")], varying = 1:6, direction = "long")
> obsCovs <- obsCovs[order(obsCovs$id, obsCovs$time), c(2:3)]
> wt <- unmarkedFrameOccu(y = y, siteCovs = siteCovs, obsCovs = obsCovs)
> summary(wt)
unmarkedFrame Object

237 sites
Maximum number of observations per site: 3
Mean number of observations per site: 2.81
Sites with at least one detection: 79

Tabulation of y observations:
   0    1 <NA>
 483  182   46

Site-level covariates:
      elev                forest               length
 Min.   :-1.436125   Min.   :-1.265e+00   Min.   :0.1823
 1st Qu.:-0.940726   1st Qu.:-9.744e-01   1st Qu.:1.4351
 Median :-0.166666   Median :-6.499e-02   Median :1.6094
 Mean   : 0.007612   Mean   : 8.798e-05   Mean   :1.5924
 3rd Qu.: 0.994425   3rd Qu.: 8.080e-01   3rd Qu.:1.7750
 Max.   : 2.434177   Max.   : 2.299e+00   Max.   :2.2407

Observation-level covariates:
      date                 ivel
 Min.   :-2.9043386   Min.   :-1.753e+00
 1st Qu.:-1.1186243   1st Qu.:-6.660e-01
 Median :-0.1186243   Median :-1.395e-01
 Mean   :-0.0002173   Mean   :-3.008e-11
 3rd Qu.: 1.3099471   3rd Qu.: 5.493e-01
 Max.   : 3.8099471   Max.   : 5.980e+00
 NA's   :42.0000000   NA's   : 4.600e+01
```

or by using the convenience function `csvToUMF`:

```
> wt <- csvToUMF(system.file("csv", "widewt.csv", package = "unmarked"),
     long = FALSE, type = "unmarkedFrameOccu")
```

If not all sites have the same numbers of observations, then manual importation of data in long format can be tricky. `csvToUMF` seemlessly handles this situation.

```
> pcru <- csvToUMF(system.file("csv", "frog2001pcru.csv",
     package = "unmarked"), long = TRUE, type = "unmarkedFrameOccu")
> summary(pcru)
unmarkedFrame Object

130 sites
Maximum number of observations per site: 3
Mean number of observations per site: 2.59
Sites with at least one detection: 96

Tabulation of y observations:
   0    1    2    3 <NA>
 197   25   28   87   53
```

```
Observation-level covariates:
 MinAfterSunset        Wind              Sky            Temperature   
 Min.   :-21.00   Min.   : 0.0000   Min.   : 0.0000   Min.   : 4.00  
 1st Qu.: 66.00   1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.:13.00  
 Median : 97.00   Median : 1.0000   Median : 0.0000   Median :17.50  
 Mean   : 97.57   Mean   : 0.8813   Mean   : 0.4837   Mean   :16.61  
 3rd Qu.:126.00   3rd Qu.: 2.0000   3rd Qu.: 1.0000   3rd Qu.:20.60  
 Max.   :228.00   Max.   : 3.0000   Max.   : 5.0000   Max.   :28.00  
 NA's   : 53.00   NA's   :53.0000   NA's   :53.0000   NA's   :53.00  
   JulianDate   
 Min.   : 72.0  
 1st Qu.: 95.0  
 Median :123.0  
 Mean   :127.4  
 3rd Qu.:159.0  
 Max.   :179.0  
 NA's   : 53.0  
```

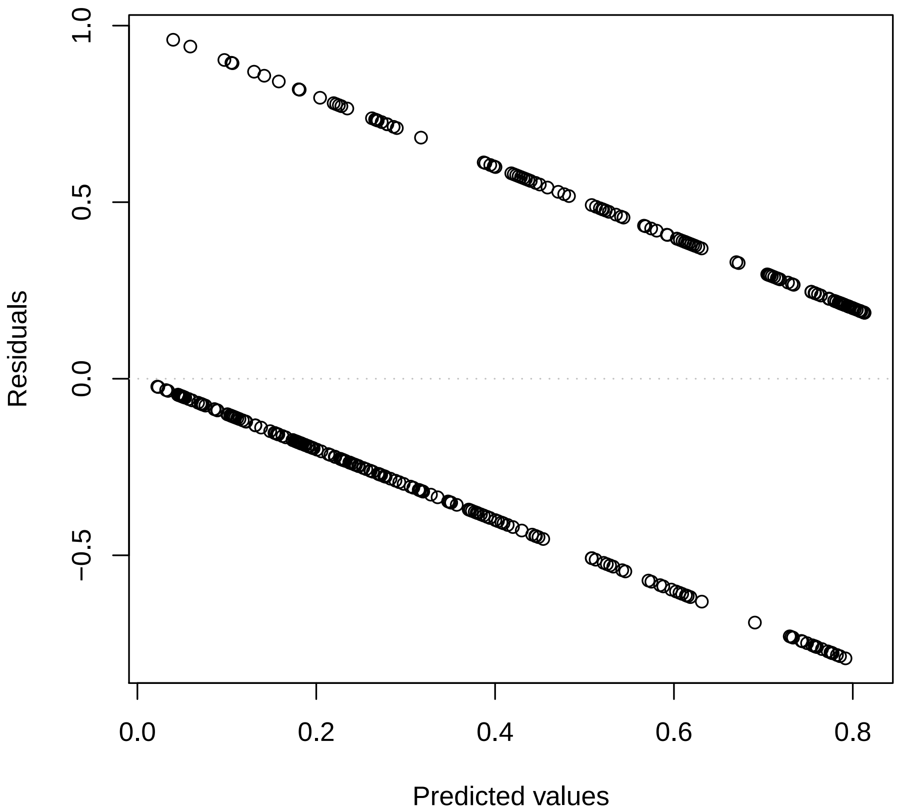
To help stabilize the numerical optimization algorithm, we recommend standardizing the covariates.

```
> obsCovs(pcru) <- scale(obsCovs(pcru))
```

Occupancy models can then be fit with the occu() function:

```
> fm1 <- occu(~1 ~ 1, pcru)
> fm2 <- occu(~MinAfterSunset + Temperature ~ 1, pcru)
> summary(fm1)
Call:
occu(formula = ~1 ~ 1, data = pcru)

Occupancy (logit-scale):
 Estimate   SE    z P(>|z|)
     2.95 1.44 2.05    0.04

Detection (logit-scale):
 Estimate    SE     z P(>|z|)
   -0.249 0.170 -1.47   0.142

AIC: 461.0986 
Sample size: 130
optim convergence code: 0
optim iterations: 22 
Bootstrap iterations: 0 
> summary(fm2)
Call:
occu(formula = ~MinAfterSunset + Temperature ~ 1, data = pcru)

Occupancy (logit-scale):
 Estimate    SE    z  P(>|z|)
     1.54 0.292 5.26 1.42e-07

Detection (logit-scale):
               Estimate    SE      z  P(>|z|)
(Intercept)      0.2098 0.206  1.017 3.09e-01
MinAfterSunset  -0.0855 0.160 -0.536 5.92e-01
Temperature     -1.8936 0.291 -6.508 7.60e-11

AIC: 357.0791 
Sample size: 130
optim convergence code: 0
optim iterations: 21 
Bootstrap iterations: 0 
```

```
> plot(fm2)
```



Here, we have specified that the detection process is modeled with the MinAfterSunset and Temperature covariates. No covariates are specified for occupancy here. See ?occu for more details.

Unmarked fitting functions return unmarkedFit objects which can be queried to investigate the model fit. Variables can be back-transformed to the unconstrained scale using backTransform. Standard errors are computed using the delta method.

```
> backTransform(fm2, "state")
Backtransformed linear combination(s) of Occupancy estimate(s)

 Estimate     SE LinComb (Intercept)
    0.823 0.0425    1.54           1

Transformation: logistic
```

Because the detection component was modeled with covariates, covariate coefficients must be specified to back-transform. Here, we request the probability of detection given a site is occupied and all covariates are set to 0.

```
> backTransform(linearComb(fm2, coefficients = c(1, 0,
     0), type = "det"))
Backtransformed linear combination(s) of Detection estimate(s)

 Estimate    SE LinComb (Intercept) MinAfterSunset Temperature
    0.552 0.051   0.210           1              0           0

Transformation: logistic
```

A predict method also exists.

```
> newData <- data.frame(MinAfterSunset = 0, Temperature = -2:2)
> predict(fm2, type = "det", newdata = newData, appendData = TRUE)
   Predicted         SE MinAfterSunset Temperature
1 0.98196076 0.01266193              0          -2
2 0.89123189 0.04248804              0          -1
3 0.55225129 0.05102660              0           0
4 0.15658708 0.03298276              0           1
5 0.02718682 0.01326263              0           2
```

Confidence intervals are requested with confint, using either the asymptotic normal approximation or profiling.

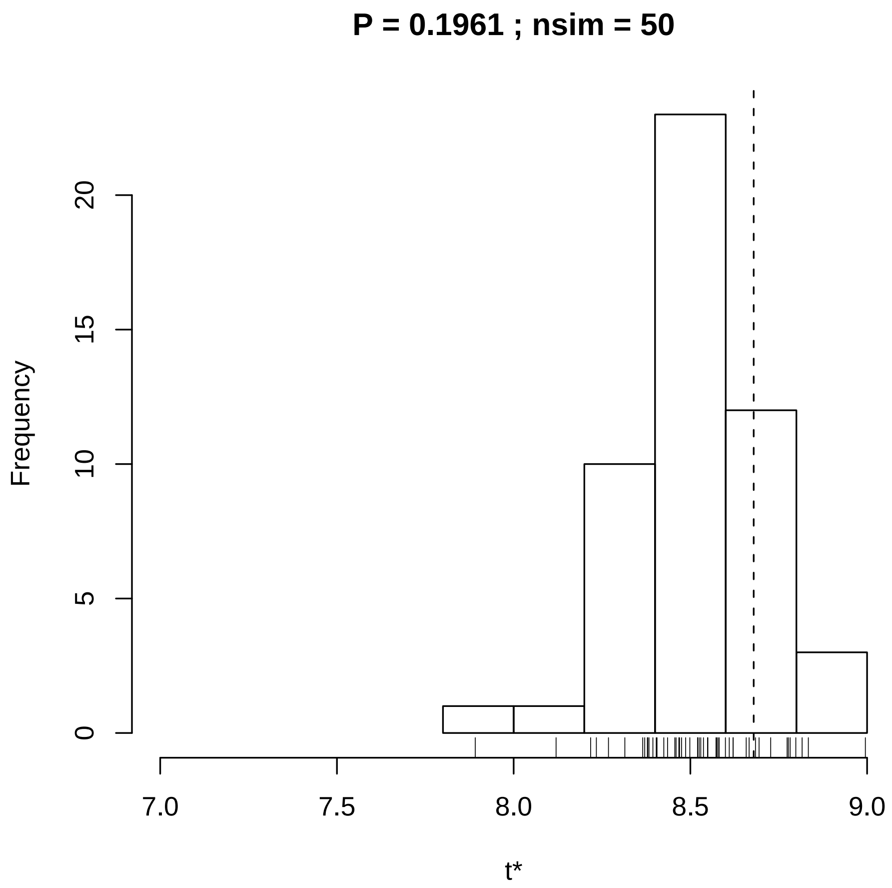
```
> confint(fm2, type = "det")
                    0.025      0.975
(Intercept)    -0.1946872  0.6142292
MinAfterSunset -0.3985642  0.2274722
Temperature    -2.4638797 -1.3233511
> confint(fm2, type = "det", method = "profile")
Profiling parameter 1 of 3 ... done.
Profiling parameter 2 of 3 ... done.
Profiling parameter 3 of 3 ... done.
                       0.025      0.975
p(Int)            -0.1929210  0.6208837
p(MinAfterSunset) -0.4044794  0.2244221
p(Temperature)    -2.5189984 -1.3789261
```

Model selection and multi-model inference can be implemented after organinzing models using the fitList function.

```
> fms <- fitList(Null = fm1, TimeTemp = fm2)
> modSel(fms, nullmod = "Null")
     model   n nPars    AIC deltaAIC      AICwt     Rsq cumltvAICwt
1 TimeTemp 130     4 357.08     0.00 1.0000e+00 0.58243           1
2     Null 130     2 461.10   104.02 2.5849e-23 0.00000           1
> predict(fms, type = "det", newdata = newData, appendData = TRUE)
   Predicted         SE MinAfterSunset Temperature
1 0.98196076 0.01266193              0          -2
2 0.89123189 0.04248804              0          -1
3 0.55225129 0.05102660              0           0
4 0.15658708 0.03298276              0           1
5 0.02718682 0.01326263              0           2
```

Parametric bootstrapping can be used to check the adequacy of model fit.

```
> pcru.pb <- parboot(fm2, nsim = 50, report = 5)
t0 = 8.679173
8.6, 9, 8.1, 8.3, 8.4
8.9, 8.4, 8.3, 8.6, 8.5
8.2, 8.5, 8.5, 8.9, 8.7
8, 8.9, 8.6, 8.8, 8.5
8.4, 8.3, 8.4, 8.8, 8.3
8.8, 7.8, 8.4, 8.6, 8.7
8.5, 8.6, 8.5, 8.6, 8.5
8.6, 8.5, 8.4, 8.3, 8
8.8, 8.6, 8.3, 8.4, 8.3
8.6, 8.6, 8.5, 8.7, 8.4
> plot(pcru.pb)
```



This example suggests an adequate fit.

References

[1] Darryl I. MacKenzie, James D. Nichols, James E. Hines, Melinda G. Knutson, and Alan B. Franklin. Estimating site occupancy, colonization, and local extinction when a species is detected imperfectly. *Ecology*, 84(8):2200–2207, 2003.

[2] Darryl I. MacKenzie, James D. Nichols, G. B. Lachman, S. Droege, J. A. Royle, and C. A. Langtimm. Estimating site occupancy rates when detection probabilities are less than one. *Ecology*, 83(8):2248–2255, 2002.

[3] J. A. Royle. Generalized estimators of avian abundance from count survey data. *Animal Biodiversity and Conservation*, 27(1):375–386, 2004.

[4] J. A Royle. N-mixture models for estimating population size from spatially replicated counts. *Biometrics*, 60(1):108–115, 2004.

[5] J. A. Royle and J. D. Nichols. Estimating abundance from repeated presence-absence data or point counts. *Ecology*, 84(3):777–790, 2003.

[6] JA Royle, DK Dawson, and S. Bates. Modeling abundance effects in distance sampling. *Ecology*, 85(6):1591–1597, 2004.