Design and Analysis of Nest Survival Studies Part 1 Using *RMark*

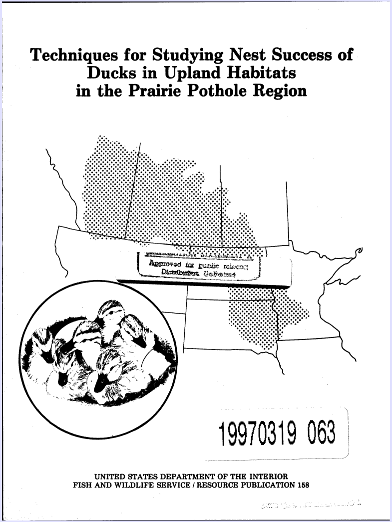
Carl James Schwarz [1]   Jim Rivers [2]

[1]StatMathComp Consulting
Port Moody, BC, Canada
cschwarz.stat.sfu.ca @ gmail.com

[2]Oregon State University
Corvalis, Oregon
Jim.Rivers @ oregonstate.edu

Outline I

Study Protocol



Techniques for Studying Nest Success of Ducks in Upland Habitats in the Prairie Pothole Region

Approved for public release; Distribution Unlimited

19970319 063

UNITED STATES DEPARTMENT OF THE INTERIOR
FISH AND WILDLIFE SERVICE / RESOURCE PUBLICATION 158

Study Protocol

- Find and mark nest locations
- Determine state of incubation (days since laid)
- Revisit nests (e.g. 4 day intervals) until either
  - Nest failure (e.g. destroyed or predated or abandoned, etc.)
  - Nest success (e.g. at least one egg hatched)

Interest lies in

$$P = P(nest\ success)$$

Bias in Apparent Nest Success

Apparent nest success is

$$\widehat{P}^{*} = \frac{n_s}{n_s + n_f}$$

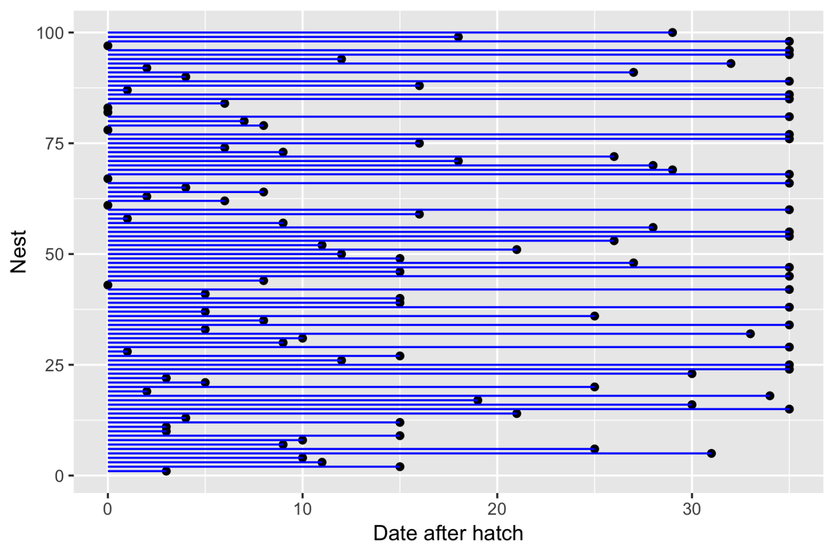
where

- n_s = observed number of success nests
- n_f = observed number of failed nests

This estimator is biased because not all nests have same probability of detection, i.e. you will miss nests that have already failed prior to your survey.

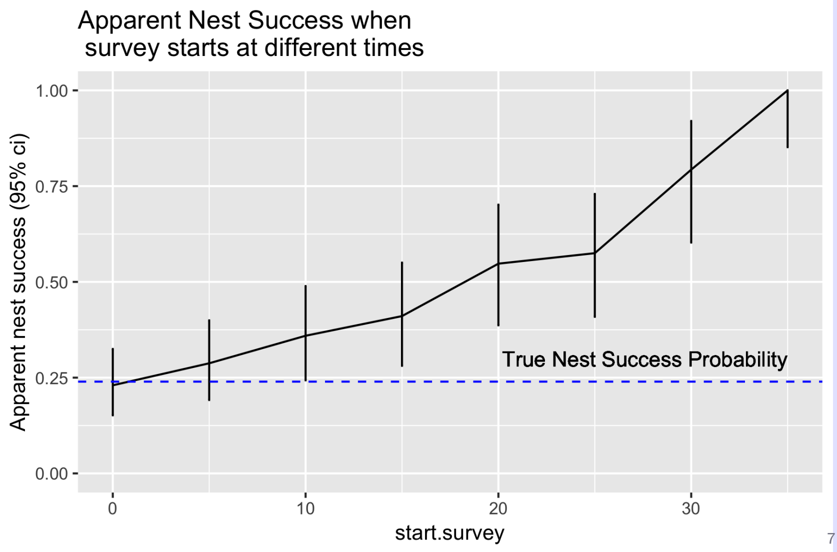
Bias in Apparent Nest Success

Simulated nest data with DSR=0.96 and hatch date = 35 days.



Bias in Apparent Nest Success

Simulated nest data with DSR=0.96 and hatch date = 35 days. Apparent Nest Success computed when survey starts at various times until hatch date.



Mayfield Method

Mayfield(1961,1975) method adjust for unequal exposure to risks of failure:

- Estimate total exposure time of observed nests
  - Use number of days between visits if nest does not fail in interval
  - Use 1/2 of number of days between visits if nest fails in interval
- $\widehat{DSR} = 1 - \frac{Nests\ \ failed}{Total\ \ Exposure\ \ Days}$
- $\widehat{p} = \widehat{DSR}^{Time\ \ to\ \ Hatch}$

An *ad hoc* method that assumes equal DSR and random failure.

Example of Mayfield Method

Consider the following study (see GIM)
Number of nests that survive to a given age:

initial sample size	*age of nest when found*	*7 days*	*14 days*	*21 days*	*28 days*	*35 days*
10	0 days	7	6	5	3	3
10	14 days			8	7	4
10	28 days					8

- Number of nests that failed $= (10-3)+(10-4)+(10-8) = 15$
- Exposure for nests in first row that survived$=$ $7+6\times 7+5\times 7+3\times 3+3 = 168$ days.
- Exposure for nests in first row that fail $=$ $3\times 3.5+1\times 3.5+1\times 3.5+2\times 3.5+0\times 3.5 = 24.5$ days
- Total exposure $= 409.5$ exposure days.
- $\widehat{DSR} = 1-\frac{15}{409.5} = .9633$
- $\widehat{P} = .9633^{35} = 0.27$

How do you estimate uncertainty? How do you relax assumption of

Moving beyond Mayfield

- Place into statistical framework
    - Model for process + maximum likelihood
    - Allow for covariates and other factors
    - AIC for model ranking; selection; prediction
- Software implementations
    - *MARK* via *RMark*; logistic-exposure models in *R*

Statistical framework

Basic statistical parameters

- $DSR_i = S_i = p(nest\ survives\ day_i)$

Focus is on the DSR rather than the P(nest survival).

Statistical framework

Important terms:

- **Population**: Complete set of nests for which inference is wanted
- **Sample**: Nest actually measured
- **Parameter**: Set of DSR_d on day d over entire population - always unknown
- **Estimate**: Values obtained from the sample data ($\widehat{DSR}_d$).

The RRR's of statistics:

- **Randomization**: Makes a sample representative.
- **Replication**: Control precision (the SE)
- **Stratification**: Account for one form of heterogeneity in population.

Statistical framework

Every SAMPLE will give a DIFFERENT estimate of DSR!

A measure of how much the estimate could vary if a new sample is taken is the STANDARD ERROR (SE).

A usual convention is to report a 95% CONFIDENCE INTERVAL which is a plausible range for the POPULATION parameter given the collected data. A c.i. does NOT make statement about individual values (i.e. the survival of a particular nest).

Statistical framework

LIKELIHOOD connects the data (nest monitoring data) with the parameters of the model (DSR). Need to construct a probability expression for each observed nest in the sample.

(Summary) date for each nest

- Time the nest was first encountered
- Last date when the nest was known to be alive,
- Last date when the nest was checked,
- Fate of nest on last date the nest was checked

For example, here is some data:

id	FirstFound	LastPresent	LastChecked	Fate
/*A*/	1	9	9	0
/*B*/	5	5	9	1
/*C*/	5	40	40	0
...				

Notice that $Fate = 1$ is a nest failure.

Statistical framework

Two cases;

- Nest not failed on last date when nest checked.
  $L_i = S_{FirstFound} S_{FirstFound+1} \ldots S_{LastChecked-1}$
- Nest failed on last date when nest checked.
  $L_i = S_{FirstFound} S_{FirstFound+1} \ldots S_{LastAlive-1} \times (1 - S_{LastAlive} S_{LastAlive+1} \ldots S_{LastChecked-1})$

Effective Sample Size I

What is the effective sample size?

Consider nest entry:

id	FirstFound	LastPresent	LastChecked	Fate
/*A*/	1	9	9	0
/*B*/	5	5	9	1
/*C*/	5	40	40	0

...

- We assume that nest fate in each day is independent of nest fate in anyother day, so a span of x days is equivalent to x individual visits to a nest. So for nest A, there are effectively 8 data values for the interval from1 to 9 days.
- For the last interval, we don't know the time of failure, so this is counted as one interval.
- So for the above 3 nests, the effective sample size is
  - Nest A: 8

Effective Sample Size II

- Nest B: 1
- Nest C: 35

Note a slight error in GIM, Chapter 17, page 17-8 where they miscount by 1.
The assumption of independent fates for each day is quite strong (!)

Statistical framework

Overall likelihood is

$$L = L_1 \times L_2 \times L_3 \times \ldots L_n$$

Find the values of the parameter (the MLEs) that MAXIMIZE the overall probability of the data. In simple cases this can be done by hand, but usually is done numerically.

The SE can also be obtained from derivatives of the (log)-likelihood function

Statistical framework

Properties of MLEs

- Extracts ALL of the information from the data
- Give the smallest possible SE
- Are unbiased (in large samples)
- Enable CI to be constructed as $Estimate \pm 2SE$.

However, in small samples, MLEs are not guaranteed to be optimal.

Statistical framework

Software to fit model

- *MARK*
- *MARK* via *RMark*
- Logistic exposure models in *R*

Both will give same answers.
Tradeoff between flexibility and ease of use.


Program MARK
You can obtain context-sensitive help with the F1 key,
and can investigate objects with the Shift-F1 key.
See the Help menu for known problems.
Colorado State
Colorado State

Using *RMark*

RMark is an *R* package that interfaces with *MARK*
Key advantages are:

- Scripts so that analyzes can be reused.
- No more clicking on PIMs
- Much easier extracting output

Key disadvantages are:

- Only available on Windoze platforms.

Installation of package

- Download and Install package file from CRAN in the usual way

Using *RMark*

Basic steps in analysis:

- Input the nest data and *process* the data.
- Define factors and covariates in the *ddl* (optional)
- Fit some model using the *mark* function.
- Extract information from returned object
- Model average using the *collect.models* function.
- Extract results and plot results etc

Analysis of killdeer data using *RMark*

Open the *killdeer.xlsx* workbook.

For example, here is some data:

id	FirstFound	LastPresent	LastChecked	Fate
/*A*/	1	9	9	0
/*B*/	5	5	9	1
/*C*/	5	40	40	0
...				

Notice that *Fate* $= 1$ is a nest failure.

In the next worksheet, the Mayfield estimate is computed.

Analysis of killdeer data using *RMark*

Open the *killdeer.R* script.

We read in the raw data.
Notice the fieldnames MUST match exactly as given. but the order of columns can differ. The *id* column is optional.

```
killdata <- readxl::read_excel("Killdeer.xlsx",
                               sheet="killdeer")
head(killdata)
```

```
> head(killdata)
  id     FirstFound LastPresent LastChecked  Fate  Freq
1 /*A*/           1           9           9     0     1
2 /*B*/           5           5           9     1     1
3 /*C*/           5          40          40     0     1
4 /*D*/           9          32          32     0     1
5 /*E*/           7           8           8     0     1
6 /*F*/           3          15          15     0
```

Analysis of killdeer data using *RMark*

What are the parameter names for this model in *RMark*?

```
setup.parameters("Nest", check=TRUE)
```

```
> setup.parameters("Nest", check=TRUE)
[1] "S
```

Analysis of killdeer data using *RMark* I

Process the data

```
kill.proc <- process.data(killdata, model="Nest", nocc=40)
kill.proc
```

nocc is the MAXIMUM number of days on which nests are located.

```
> kill.proc
$data
# A tibble: 18 x 6
   id     FirstFound LastPresent LastChecked  Fate  freq
   <chr>       <dbl>       <dbl>       <dbl> <dbl> <dbl>
 1 /*A*/           1           9           9     0     1

$model
[1] "Nest"

$mixtures
```

Analysis of killdeer data using *RMark* II

```
[1] 1

$freq
   group1
1       1
2       1
...
```

Examine the *ddl* and modify as needed (refer to later programs)

```
# 2. Examine and/or modify the ddl. (Not done here)
kill.ddl <- make.design.data(kill.proc)
kill.ddl
```

```
> # 2. Examine and/or modify the ddl. (Not done here)
> kill.ddl <- make.design.data(kill.proc)
> str(kill.ddl)
List of 2
 $ S          :'data.frame': 39 obs. of  7 variables:
  ..$ par.index  : int [1:39] 1 2 3 4 5 6 7 8 9 10 ...
  ..$ model.index: num [1:39] 1 2 3 4 5 6 7 8 9 10 ...
  ..$ group      : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1
  ..$ age        : Factor w/ 39 levels "0","1","2","3",..:
  ..$ time       : Factor w/ 39 levels "1","2","3","4",..:
  ..$ Age        : num [1:39] 0 1 2 3 4 5 6 7 8 9 ...
```

Analysis of killdeer data using *RMark* II

```
  ..$ Time        : num [1:39] 0 1 2 3 4 5 6 7 8 9 ...

> kill.ddl
$S
   par.index model.index group age time Age Time
1          1           1     1   0    1   0    0
2          2           2     1   1    2   1    1
3          3           3     1   2    3   2    2
4          4           4     1   3    4   3    3
....
```

Notice the difference between *time* and *Time*, etc.

Fit a model using the *mark()* function.
You need to specify a model for *S* using the usual *R* model notation.
Here we assume the *DSR* is constant over all days (similar to the Mayfield) assumption

```
mod.res <-  RMark::mark(kill.proc,
                        model="Nest",
                        model.parameters=list(
                          S   =list(formula=~1)
                        )
  )
summary(mod.Sdot)
```

```
Output summary for Nest model
Name : S(~1)

Npar :  1
-2lnL:  42.51028
AICc :  44.52951

Beta
                 estimate        se      lcl      ucl
S:(Intercept) 3.557002 0.4141776 2.745214 4.368791


Real Parameter S
         1         2         3         4         5
 0.9722669 0.9722669 0.9722669 0.9722669 0.9722669 0.972266
        15        16        17        18        19        2
```

```
0.9722669 0.9722669 0.9722669 0.9722669 0.9722669 0.972266
       29        30        31        32        33         3
0.9722669 0.9722669 0.9722669 0.9722669 0.9722669 0.972266
```

Review the returned summary.

Analysis of killdeer data using *RMark* I

Extract the individual pieces of the model fit.

```
# Look the objects returned in more details
names(mod.res)
names(mod.res$results)
```

Look at some of the individual pieces that are returned.
Many of these will become clearer when we fit more complex models.

```
> # Look the objects returned in more details
> names(mod.res)
 [1] "data"                "model"               "title"
 [8] "parameters"          "time.intervals"      "number.of.grou
[15] "fixed"               "design.matrix"       "pims"
[22] "simplify"            "model.parameters"    "results"
> names(mod.res$results)
 [1] "lnl"                 "deviance"            "deviance.df"
```

```
 [8] "real"                  "beta.vcv"             "derived"
>
```

Analysis of killdeer data using *RMark* I

Beta, Real, Derived estimates

```
# look at estimates on beta and original scale
mod.res$results$beta  # on the logit scale

mod.res$results$real# on the regular 0-1 scale for each si

# derived variabldes is the nest survival probability over
names(mod.res$results$derived)

mod.res$results$derived$"S Overall Survival"
```

Analysis of killdeer data using *RMark* II

```
> # look at estimates on beta and original scale
> mod.res$results$beta  # on the logit scale
                estimate        se      lcl      ucl
S:(Intercept) 3.557002 0.4141776 2.745214 4.368791
>
> mod.res$results$real# on the regular 0-1 scale for each s
              estimate         se       lcl        ucl fixed
S g1 a0 t1 0.9722669 0.0111679 0.9396425 0.9874919
>
> # derived variabldes is the nest survival probability ove
> names(mod.res$results$derived)
[1] "S Overall Survival"
>
> mod.res$results$derived$"S Overall Survival"
   estimate        se       lcl       ucl
1 0.3339134 0.1495833 0.1182906 0.6519547
```

Analysis of killdeer data using *RMark* III

```
>
```

Alternative ways to get the estimates

```
# alternatively
get.real(mod.res, "S", se=TRUE)

# Notice that the nest survival is the product of the indi
prod(get.real(mod.res, "S", se=TRUE)$estimate)
```

```
> # alternatively
> get.real(mod.res, "S", se=TRUE)
              all.diff.index par.index  estimate        se
S g1 a0 t1                 1         1 0.9722669 0.0111679 (
S g1 a1 t2                 2         1 0.9722669 0.0111679 (
S g1 a2 t3                 3         1 0.9722669 0.0111679 (
.....
>
> # Notice that the nest survival is the product of the inc
```

Analysis of killdeer data using *RMark* II

```
> prod(get.real(mod.res, "S", se=TRUE)$estimate)
[1] 0.3339139
```

Analysis of killdeer data using *RMark* I

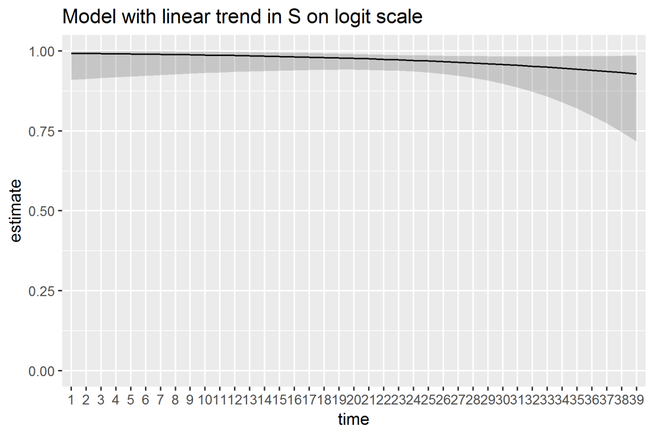
Exercises

1. Fit a linear trend to S using the *Time* variable in the *ddl*. Plot the estimates over time.
2. FIi a model with S varying in first and second half of the study
    - Create new variable in the *ddl* for 1st and 2nd half of study first.

Exercise ... Linear trend

```
# This is a model with S linear over time.
# Notice the use of Time vs time.
mod.res <-  RMark::mark(kill.proc,
                                 model="Nest",
                                 model.parameters=list(
                                   S   =list(formula=~Time)
                                 )
   )
```

Exercise ... Linear trend



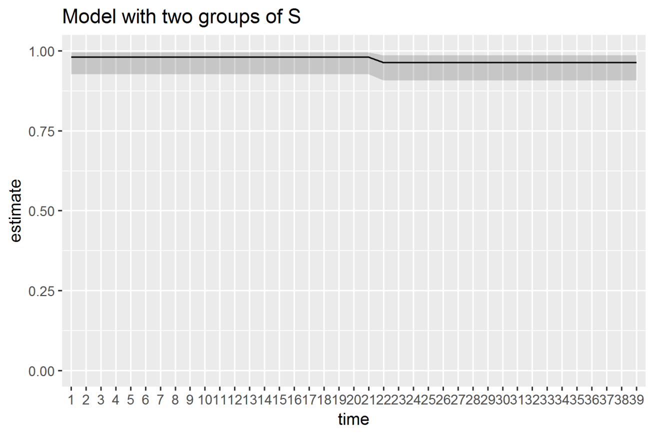
Why is the fit curved?

Analysis of killdeer data using *RMark* I

Exercise ... 2 groups of S

```
kill.ddl <- make.design.data(kill.proc)

kill.ddl$S$studyhalf <- car::recode(kill.ddl$S$Time,
                    " lo:20='1st'; 21:hi='2nd'", as.factor=T
str(kill.ddl)
kill.ddl

# 3. Fit a particular model
# This is a model with S linear over time.
# Notice the use of Time vs time.
mod.res <-  RMark::mark(kill.proc, ddl=kill.ddl,
                           model="Nest",
                           model.parameters=list(
                             S   =list(formula=~studyhalf)
                           )
  )
```

Exercise ... 2 groups of S



Analysis of killdeer data using *RMark*

Which model is best?

"All models are wrong, but some are useful" (George Box)

AIC = Akaike Information Criteria = tradeoff between fit and complexity

$$AIC = -2 \log likelihood + 2\#p$$

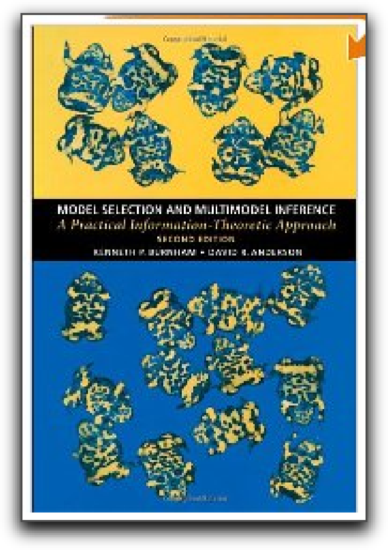
where $\#p$ is the number of parameters.

"Best" model in the set is one with arithmetically smallest AIC. Find

$$\Delta AIC_m = AIC_m - AIC_{min}$$

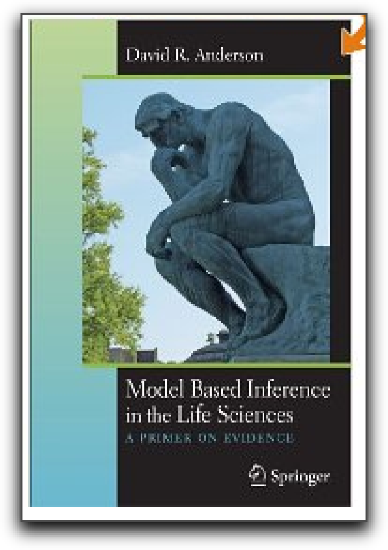
Differences of 2 or 3 AIC units imply no real preference.

Analysis of killdeer data using *RMark*



Burnham, K.P. and Anderson, D.R. (2002).
Model selection and Multi-Model Inference: A practical information-theoretic approach.

Complete but dense. The first few chapters are very readable.



Anderson, D.R. (2008).
Model based inference in the life sciences: A primer on evidence.

Very readable explanation of the AIC paradigm.

Model averaging - compute a model weight

$$w_m = \frac{\exp(-\frac{\Delta AIC_m}{2})}{\sum \exp(-\frac{\Delta AIC_i}{2})}$$

Interpret as the weight of evidence in favor of model *m* relative to OTHER MODELS in the model set.

Add up weights for models with a common feature, e.g. habitat variable (see later in notes).

Model averaging - average estimates

$$DSR_{average} = \sum w_m DSR_m$$

This accounts for both WITHIN and AMONG model uncertainty.

Also able to obtain SE that are adjusted for model averaging (weighted average of the individual standard errors and a penalty if the estimates are not consistent among models).

Analysis of killdeer data using *RMark* I

Open the *killdeer-modavg.r* script.

We will compare the model with constant S, linear S, and quadratic S.

Data read in the usual way.
Add a variable to the *ddl* for the quadratic term

```
# 2. Examine and/or modify the ddl. Here you could standar
kill.ddl <- make.design.data(kill.proc)
kill.ddl$S$Time2 <- (kill.ddl$S$Time-20)^2
kill.ddl
```

```
$S
   par.index model.index group age time Age Time Time2
1          1           1     1   0    1   0    0   400
2          2           2     1   1    2   1    1   361
3          3           3     1   2    3   2    2   324
4          4           4     1   3    4   3    3   289
...
```

Analysis of killdeer data using *RMark* I

Create the model set. Order of the potential models is not important

```
# 3. Set up the set of model to fit
model.list.csv <- textConnection(
" S
 ~1
 ~Time
 ~Time+Time2
")

model.list <- read.csv(model.list.csv, header=TRUE, as.is=
model.list$model.number <- 1:nrow(model.list)
model.list
```

Analysis of killdeer data using *RMark* II

```
> model.list
              S model.number
1            ~1            1
2         ~Time            2
3 ~Time+Time2            3
```

Analysis of killdeer data using *RMark* I

Fit all of the models in the model set:

```
model.fits <- plyr::dlply(model.list, "model.number", funct
  cat("\n\n***** Starting ", unlist(x), "\n")

  fit <- RMark::mark(input.data, ddl=input.ddl,
                    model="Nest",
                    model.parameters=list(
                      S    =list(formula=as.formula(eval(x$
                    )
                    #,brief=TRUE,output=FALSE, delete=TRU
                    #,invisible=TRUE,output=TRUE  # set fo
  )
  mnumber <- paste("m...",formatC(x$model.number, width = 3
  assign( mnumber, fit, envir=.GlobalEnv)
  #browser()
  fit

},input.data=kill.proc, input.ddl=kill.ddl)
```

Analysis of killdeer data using *RMark* II

RMark succcessively calls *MARK* and stores the results in *m..001* etc.

Analysis of killdeer data using *RMark* I

Collect the models and compute the AIC table.

```
# Model comparision and averaging
# collect models and make AICc table

model.set <- RMark::collect.models( type="Nest")
model.set
```

```
> model.set
              model npar     AICc DeltaAICc    weight Deviar
3 S(~Time + Time2)    3 44.19824 0.0000000 0.3992023 38.08
1             S(~1)    1 44.52951 0.3312659 0.3382669 42.51(
2          S(~Time)    2 45.03644 0.8381992 0.2625309 40.978
```

Interpret the table

Analysis of killdeer data using *RMark* I

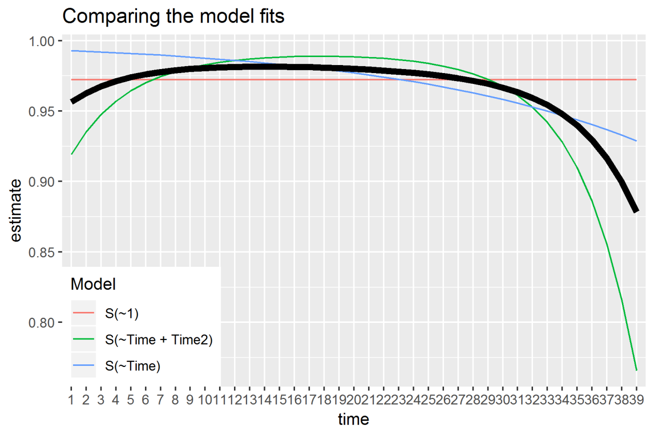
Get model average value of DSR.

```
S.ma <- RMark::model.average(model.set, param="S")
S.ma
```

The individual DSR values are model averaged:

```
> S.ma
                par.index  estimate          se fixed     note g
S g1 a0 t1              1 0.9565048 0.07681821
S g1 a1 t2              2 0.9627822 0.05732738
S g1 a2 t3              3 0.9675655 0.04313985
S g1 a3 t4              4 0.9712112 0.03293324
S g1 a4 t5              5 0.9739912 0.02569978
S g1 a5 t6              6 0.9761105 0.02068439
....
```

Plotting the individual curves and the model averaged curve.



Analysis of killdeer data using *RMark*

It is also possible to model average the derived parameters. Note that *RMark* doesn't have a model averaging function for derived parameters, so use my code.

```
source("RMark.additional.functions.r")
...
RMark.model.average.derived(model.set, param="S Overall Su
```

```
> RMark.model.average.derived(model.set, param="S Overall S
   estimate        se        lcl       ucl
1 0.2648794 0.1701028 0.06105668 0.6662856
```

Analysis of killdeer data using *RMark*

Exercise

1. Add the 1st and 2nd half model to the model set

Exercise - Sherry - 1

Articles

Editor's Choice

Journal of Avian Biology 46: 559–569, 2015
doi: 10.1111/jav.00536

Subject Editor: Peter Arcese. Editor-in-Chief: Thomas Alerstam. Accepted 5 April 2015

Impacts of nest predators and weather on reproductive success and population limitation in a long-distance migratory songbird

Thomas W. Sherry, Scott Wilson, Sarah Hunter and Richard T. Holmes

Exercise - Sherry - 1

1. Refer to first paragraph under Results - Experimental nest protection.
    - Find DSR for control nests and nests with baffles separately.
    - Estimate nest success for 20 days - need to use *delta.method.special()* function. (see code)
2. Refer to second paragraph under Results - Annual and seasonal effects.
   Fit linear and quadratic effect of date (relative to 27 May).
3. Compare constant, linear, quadratic trends in DSR using model averaging.

Covariates

Often the influence of covariates on the DSR is of interest.

Covariates can be:

- **Categorical** e.g. habitat type
- **Continuous**, e.g. distance from water

Covariates can operate at the

- **Nest level** are are fixed for the duration of the nest, e.g. distance from water
- **Day level** and are common to all nests, e.g. linear trend in DSR
- **Nest x Day** level where each nest's covariates vary over the days, e.g. nest-age, mowing

The **Nest x Day** covariates are tedious to implement (at best) in *MARK* and *RMark*.

Covariates

Hypotheses about covariates

- Is there evidence of an effect? Look at estimates/se and model selection table
- Estimate DSR at levels of covariates

Covariates

Nest-level covariates.

- Continuous covariates
  - Enter as a numeric columns in the nest data frame.
  - Specify variable name in formula, e.g. $S = list(formula = \sim Distance)$.
- Categorical covariates
  - Enter as an alphanumeric columns in the nest data frame and declare as a factor.
  - Specify the categorical covariates in the *groups=* option in the *process.data()* function.
  - Specify variable name in formula, e.g. $S = list(formula = \sim Treatment)$.

Nest level categorical covariates I

Read in the **mallard** dataset.

```
malldata <- readxl::read_excel("mallard.xlsx",
                               sheet="mallard")
head(malldata)
malldata <- as.data.frame(malldata) #avoid tibble problems


malldata$Habitat <- factor(malldata$Habitat)  # RMark wants
```

RMark sometimes gets mixed up with tibbles so convert to data frame.
RMark prefers that categorical variables be declared as factors.

Nest level categorical covariates I

When processing the data define the **groups** defined by nest level categorical variables.

```
mall.proc <- process.data(malldata, model="Nest",
                          group="Habitat",
                          nocc=max(mallard$LastChecked))
mall.proc
```

RMark converts to series of indicator variables.

```
mall.proc$freq
    HabitatN HabitatP HabitatR HabitatW
1          0        0        1        0
2          0        1        0        0
3          0        1        0        0
4          1        0        0        0
...
```

Nest level categorical covariates II

```
$group.covariates
  Habitat
1       N
2       P
3       R
4       W
```

Nest level categorical covariates I

Use the categorical variable in the model

```
mod.res <-  RMark::mark(mall.proc, ddl=mall.ddl,
                              model="Nest",
                              model.parameters=list(
                                S    =list(formula=~Habitat)
                              )
  )
```

```
> summary(mod.res)
Output summary for Nest model
Name : S(~Habitat)

Npar :  4
-2lnL:  1563.951
AICc :  1571.957
```

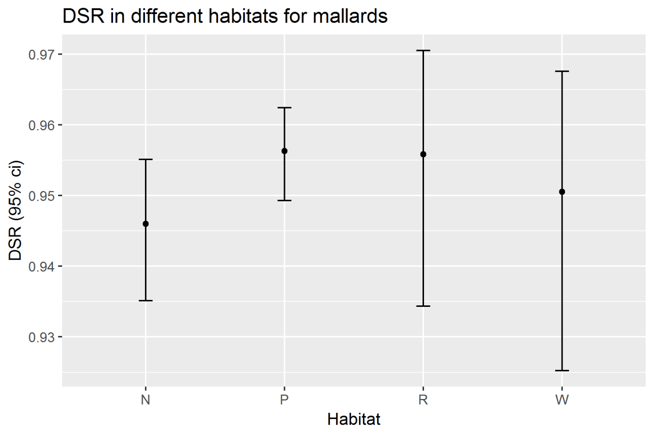
Nest level categorical covariates II

```
Beta
                 estimate        se        lcl       ucl
S:(Intercept) 2.8629313 0.0992682  2.6683656 3.0574970
S:HabitatP    0.2226790 0.1273492 -0.0269255 0.4722835
S:HabitatR    0.2111142 0.2356356 -0.2507317 0.6729600
S:HabitatW    0.0929137 0.2454492 -0.3881667 0.5739941



Real Parameter S
                        1         2         3         4
Group:HabitatN 0.9459833 0.9459833 0.9459833 0.9459833 0.94
Group:HabitatP 0.9562953 0.9562953 0.9562953 0.9562953 0.95
Group:HabitatR 0.9558094 0.9558094 0.9558094 0.9558094 0.95
Group:HabitatW 0.9505390 0.9505390 0.9505390 0.9505390 0.95
```

Gives estimates of DSR for each day in each habitat.

Nest level categorical covariates

These can be extracted and plotted (see code)



Nest level categorical covariates I

"Testing" for covariate effects (standard null hypothesis testing) is NOT recommended as does not provide useful information.

Better to get estimates use AIC with a null model to see the weight of evidence, followed by model averaging.

Fit a null model and do AIC.

```
mod.null <-  RMark::mark(mall.proc, ddl=mall.ddl,
                         model="Nest",
                         model.parameters=list(
                           S   =list(formula=~1)
                         )
)
summary(mod.null)

collect.models(type="Nest")
```

```
> collect.models(type="Nest")
        model npar     AICc DeltaAICc   weight Deviance
2       S(~1)    1 1569.117  0.000000 0.805384 1567.116
1 S(~Habitat)    4 1571.957  2.840582 0.194616 1563.951
```

Not much evidence for an impact of habitat on the DSR relative to the null model.

Nest level continuous covariates I

Use the continuous variable in the model directly.
You may wish to standardize covariates that take large values.

```
mod.rob <-  RMark::mark(mall.proc, ddl=mall.ddl,
                              model="Nest",
                              model.parameters=list(
                                S    =list(formula=~Robel)
                              )
  )
summary(mod.rob)
```

```
> summary(mod.rob)
Output summary for Nest model
Name : S(~Robel)

Npar :  2
-2lnL:  1566.773
```

Nest level continuous covariates II

```
AICc :  1570.775

Beta
                 estimate        se        lcl       ucl
S:(Intercept) 2.9088384 0.1744305  2.5669545 3.2507222
S:Robel       0.0272703 0.0466152 -0.0640954 0.1186361



Real Parameter S
        1        2        3        4        5        6
 0.952906 0.952906 0.952906 0.952906 0.952906 0.952906 0.95
       16       17       18       19       20       21
 0.952906 0.952906 0.952906 0.952906 0.952906 0.952906 0.95
```

Estimated slope (on logit scale) is .027 (SE .047) and 95% ci for slope includes zero.
Estimates of DSR are for AVERAGE value of covariate.

Nest level continuous covariates I

By default, estimates of DSR are taken at average level of covariates.
To get individual estimates, use *covariate.predictions()*.

```
# Which parameter do you want to predict
# we need to use covariate predictions to get estimated DS
head(get.real(mod.rob, "S", se=TRUE))
# because the DSR depends on the Robel value and NOT the d
# corresponding to index.all.diff=1
```

```
> # we need to use covariate predictions to get estimated
> head(get.real(mod.rob, "S", se=TRUE))
            all.diff.index par.index estimate        se
S g1 a0 t1               1         1 0.952906 0.0025942 0.9
S g1 a1 t2               2         1 0.952906 0.0025942 0.9
S g1 a2 t3               3         1 0.952906 0.0025942 0.9
S g1 a3 t4               4         1 0.952906 0.0025942 0.9
```

Nest level continuous covariates II

```
S g1 a4 t5                5           1 0.952906 0.0025942 0.94
S g1 a5 t6                6           1 0.952906 0.0025942 0.94
> # because the DSR depends on the Robel value and NOT the
> # corresonding to all.diff.index=1
>
```

THIS IS A BIT TRICKY so review carefully.

Nest level continuous covariates I

Create list of value of covariates for which you want predictions and then get predictions.

```
range(malldata$Robel)
pred.data <- data.frame(Robel=seq(min(malldata$Robel), max(
                                 index=1)
head(pred.data)

# we plot the results
# because the DSR is the same for all days, we use the time
plotdata <- covariate.predictions(mod.rob, data=pred.data)$
head(plotdata)
```

```
> range(malldata$Robel)
[1] 0.625 9.250
> pred.data <- data.frame(Robel=seq(min(malldata$Robel), ma
+                                 index=1)
> head(pred.data)
      Robel index
1 0.6250000     1
2 0.8010204     1
3 0.9770408     1
4 1.1530612     1
5 1.3290816     1
6 1.5051020     1
>
> # we plot the results
> # because the DSR is the same for all days, we use the ti
> plotdata <- covariate.predictions(mod.rob, data=pred.data
```

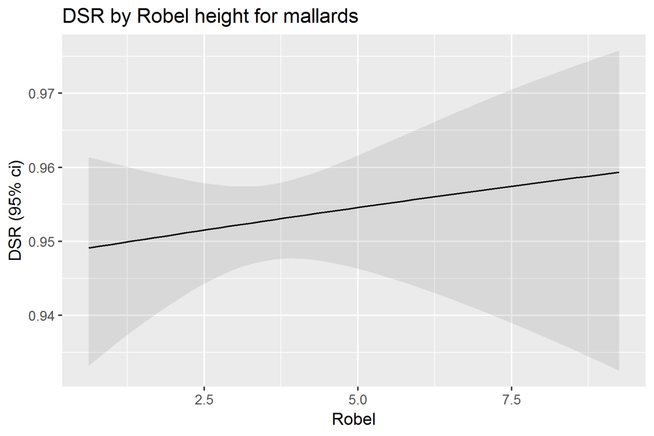
Nest level continuous covariates III

```
> head(plotdata)
  vcv.index model.index par.index     Robel index  estimate
1         1           1         1 0.6250000     1 0.949111
2         1           1         1 0.8010204     1 0.949342
3         1           1         1 0.9770408     1 0.949572
4         1           1         1 1.1530612     1 0.949802
5         1           1         1 1.3290816     1 0.950030
6         1           1         1 1.5051020     1 0.950258
```

Which can now be plotted

Nest level continuous covariates

These can be extracted and plotted (see code)



Nest level continuous covariates I

“Testing” for covariate effects (standard null hypothesis testing) is NOT recommended as does not provide useful information.

Better to get estimates use AIC with a null model to see the weight of evidence, followed by model averaging.

Fit a null model and do AIC.

```
mod.null <-  RMark::mark(mall.proc, ddl=mall.ddl,
                           model="Nest",
                           model.parameters=list(
                             S    =list(formula=~1)
                           )
)
summary(mod.null)

collect.models(type="Nest")
```

```
 collect.models(type="Nest")
      model npar     AICc DeltaAICc    weight Deviance
1     S(~1)    1 1569.117  0.000000 0.6961759 1567.116
2 S(~Robel)    2 1570.775  1.658307 0.3038241 1566.773
```

Not much evidence for an impact of Robel height on the DSR relative to the null model.

Sampling occasion covariates

These covariates apply to the sample occasions for all nests.

Add these to the *ddl*.

You already did this when you fit the linear and quadratic relationship of DSR with date of sampling.

Not discussed further here.

Exercise - Sherry - 2

Journal of Avian Biology 46: 559–569, 2015
doi: 10.1111/jav.00536

Subject Editor: Peter Arcese. Editor-in-Chief: Thomas Alerstam. Accepted 5 April 2015

Articles

Editor's Choice

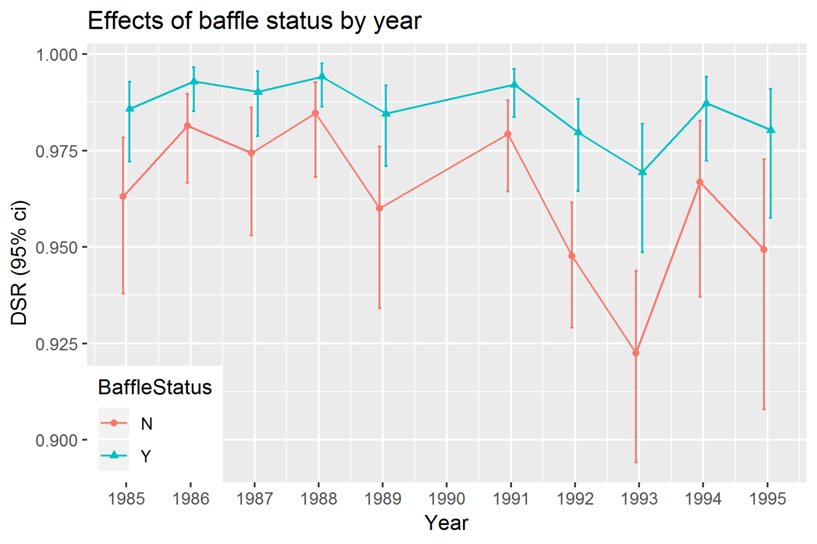
Impacts of nest predators and weather on reproductive success and population limitation in a long-distance migratory songbird

Thomas W. Sherry, Scott Wilson, Sarah Hunter and Richard T. Holmes

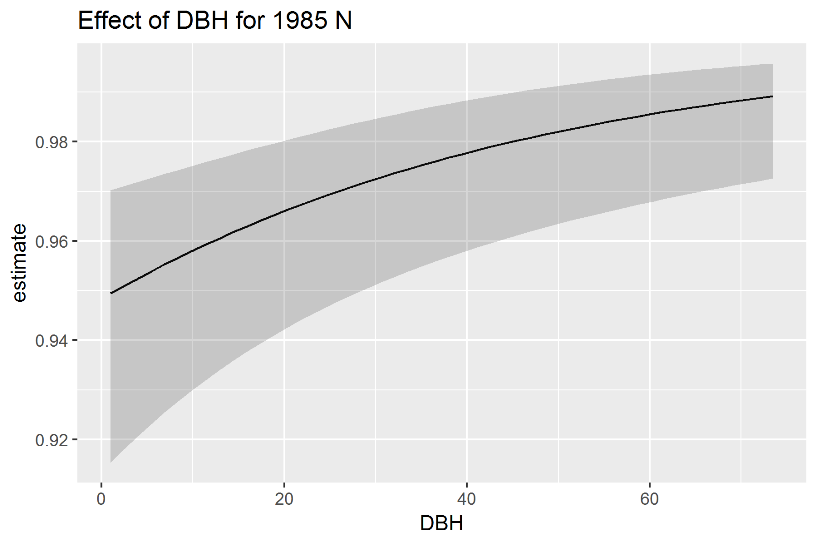
Exercise - Sherry - 2

1. Refer to Table 1a.
   - Reproduce Table 1a (use the model averaging example from killdeer).
   - Create a graphic for the model averaged results of Table 1a.
   - Plot the DSR by DBH for one year & baffle status

Exercise - Sherry - 2



Exercise - Sherry - 2



Nest x Time covariates

These covariates vary by nest for each day of the study. These are unlikely to be used in nest studies except for **Nest Age**.

In general, if you have 60 days in your study, each nest will need columns of data for each day of the study, i.e. an $n \times D$ matrix of values!

This get even more unwieldly if the covariate is categorical with k categories because you need to create $k-1$ indicator columns for each day of the study resulting in a $n \times D \times k - 1$ matrix of values!

However, *RMark* and *MARK* have a shortcut for **Nest Age**.

Age effects

RMark used the variable **AgeDay1** for the age of the nest on the first day of the season.

> *"Because the age variable is for the first day of the nesting season, this value will be negative for all the nests in the sample except those that were initiated (started) on or before day 1."*
>
> *" The age variable as described above might seem a bit odd at first glance. Why do we want to work with the age of each nest on the first date of the nesting season? The short answer is it allows us to generate each nests age on every other day of the nesting season and these are critical to have if we think that DSR might vary by nest age. "*

The variable **AgeDay1** in the datafile.which generates a variable **NestAge** for every day for every nest that can be used in the modelling.

Including age effects

Open the *killdeer.xlsx* workbook.

For example, here is some data in the *killdeer-age* worksheet:

id	FirstFound	LastPresent	LastChecked	Fate	AgeDay1
/*A*/	1	9	9	0	1
/*B*/	5	5	9	1	-2
/*C*/	5	40	40	0	-3

...

Notice that *Fate* $= 1$ is a nest failure.

Interpret the **AgeDay1** variable.

Analysis of killdeer data using *RMark*

Open the *killdeer-age.R* script.

We read in the raw data.
Notice the fieldnames MUST match exactly as given. but the order of columns can differ. The *id* column is optional.

```
killdata <- readxl::read_excel("Killdeer.xlsx",
                               sheet="killdeer-age")
head(killdata)
```

```
> head(killdata)
     id FirstFound LastPresent LastChecked Fate Freq AgeDay1
1 /*A*/          1           9           9    0    1
2 /*B*/          5           5           9    1    1       -
3 /*C*/          5          40          40    0    1       -
4 /*D*/          9          32          32    0    1       -
5 /*E*/          7           8           8    0    1       -
6 /*F*/          3          15          15    0    1
```

Analysis of killdeer data using *RMark*

What are the parameter names for this model in *RMark*?

```
setup.parameters("Nest", check=TRUE)
```

```
> setup.parameters("Nest", check=TRUE)
[1] "S
```

Analysis of killdeer data using *RMark* I

Process the data

```
kill.proc <- process.data(killdata, model="Nest", nocc=40)
kill.proc
```

nocc is the MAXIMUM number of days on which nests are located.

```
> kill.proc
$data
   id     FirstFound LastPresent LastChecked  Fate  freq
 1 /*A*/           1           9           9     0     1

$model
[1] "Nest"

$mixtures
[1] 1
```

```
$freq
   group1
1       1
2       1
...
```

Analysis of killdeer data using *RMark* I

Examine the *ddl* and modify as needed (refer to later programs)

```
# 2. Examine and/or modify the ddl. (Not done here)
kill.ddl <- make.design.data(kill.proc)
kill.ddl
```

```
> # 2. Examine and/or modify the ddl. (Not done here)
> kill.ddl <- make.design.data(kill.proc)
> str(kill.ddl)
List of 2
 $ S        :'data.frame': 39 obs. of  7 variables:
  ..$ par.index  : int [1:39] 1 2 3 4 5 6 7 8 9 10 ...
  ..$ model.index: num [1:39] 1 2 3 4 5 6 7 8 9 10 ...
  ..$ group      : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1
  ..$ age        : Factor w/ 39 levels "0","1","2","3",..:
  ..$ time       : Factor w/ 39 levels "1","2","3","4",..:
  ..$ Age        : num [1:39] 0 1 2 3 4 5 6 7 8 9 ...
```

Analysis of killdeer data using *RMark* II

```
  ..$ Time         : num [1:39] 0 1 2 3 4 5 6 7 8 9 ...

> kill.ddl
$S
   par.index model.index group age time Age Time
1          1           1     1   0    1   0    0
2          2           2     1   1    2   1    1
3          3           3     1   2    3   2    2
4          4           4     1   3    4   3    3
....
```

Notice the difference between *time* and *Time*, etc.

Analysis of killdeer data using *RMark* I

Fit a model using the *mark()* function.
You need to specify a model for S using the usual R model notation.
Here we model *DSR* as a function of nest age

```
mod.res <-  RMark::mark(kill.proc,
                             model="Nest",
                             model.parameters=list(
                               S   =list(formula=~NestAge)
                             )
  )
summary(mod.Sdot)
```

Analysis of killdeer data using *RMark* II

```
> summary(mod.res)
Output summary for Nest model
Name : S(~NestAge)

Npar :  2
-2lnL:  42.38081
AICc :  46.43878

Beta
                 estimate        se        lcl       ucl
S:(Intercept)   3.7952906 0.7988475  2.2295494 5.3610318
S:NestAge      -0.0188125 0.0516282 -0.1200038 0.0823789


Real Parameter S
         1         2         3         4         5
```

```
 0.9813396 0.9809919 0.9806379 0.9802775 0.9799105 0.979536
        15        16      17        18        19        20
 0.9758527 0.9754054 0.97495 0.9744864 0.9740145 0.9735341
        29        30        31        32        33        34
 0.9688037 0.9682301 0.9676463 0.9670521 0.9664474 0.965832
```

CAUTION: The reported estimates are the estimated DSR on day 1 at the average value of nest age across the observed nests.

Beta, Real, Derived estimates

```
# look at estimates on beta and original scale
mod.res$results$beta  # on the logit scale

mod.res$results$real# on the regular 0-1 scale for each si
```

```
> mod.res$results$beta  # on the logit scale
                 estimate        se        lcl       ucl
S:(Intercept)  3.7952906 0.7988475  2.2295494 5.3610318
S:NestAge     -0.0188125 0.0516282 -0.1200038 0.0823789
>
> mod.res$results$real# on the regular 0-1 scale for each c
               estimate        se       lcl       ucl fixed
S g1 a0 t1    0.9813396 0.0222450 0.8294276 0.9982449
S g1 a1 t2    0.9809919 0.0217489 0.8399191 0.9980340
S g1 a2 t3    0.9806379 0.0212325 0.8498100 0.9977991
```

Analysis of killdeer data using *RMark* II

```
S g1 a3 t4   0.9802775 0.0206960 0.8591143 0.9975377
S g1 a4 t5   0.9799105 0.0201399 0.8678479 0.9972474
```

CAUTION: The reported real estimates are the estimated DSR on day 1 at the average value of nest age across the observed nests.

CAUTION: The reported real estimates are the estimated DSR on day 1 at the average value of nest age across the observed nests.

```
# The real estimates are not useful because the value for
# nest ages at that time.
# For example, the beta values are shown above and the
# logit(DSR) for nest 1 day old is
logit_DSR_1 = sum( mod.res$results$beta$estimate *c(1,1))
logit_DSR_1
# and estimate of survival of nest 1 day old is
1/(1+exp(-logit_DSR_1))

# compared to
head(mod.res$results$real)

# The average nest age at day 1 is
average_nest_age_1 = mean(killdata$AgeDay1)
average_nest_age_1
logit_DSR_1_avg = sum( mod.res$results$beta$estimate *c(1,
```

Analysis of killdeer data with nest age II

```
logit_DSR_1_avg
# and estimate of survival of nest on day 1 at average age
1/(1+exp(-logit_DSR_1_avg))  # now matches the real estima
```

```
> logit_DSR_1 = sum( mod.res$results$beta$estimate *c(1,1)
> logit_DSR_1
[1] 3.776478
> # and estimate of survival of nest 1 day old is
> 1/(1+exp(-logit_DSR_1))
[1] 0.9776096
>
> # compared to
> head(mod.res$results$real)
             estimate        se       lcl       ucl fixed
S g1 a0 t1 0.9813396 0.0222450 0.8294276 0.9982449
>
```

```
> # The average nest age at day 1 is
> average_nest_age_1 = mean(killdata$AgeDay1)
> average_nest_age_1
[1] -8.888889
> logit_DSR_1_avg = sum( mod.res$results$beta$estimate *c(1
> logit_DSR_1_avg
[1] 3.962513
> # and estimate of survival of nest on day 1 at average ag
> 1/(1+exp(-logit_DSR_1_avg))  # now matches the real estin
[1] 0.9813396
```

Analysis of killdeer data with nest age I

We need to use *covariate.predictions()* to get the actual relationship of DSR and nest age.

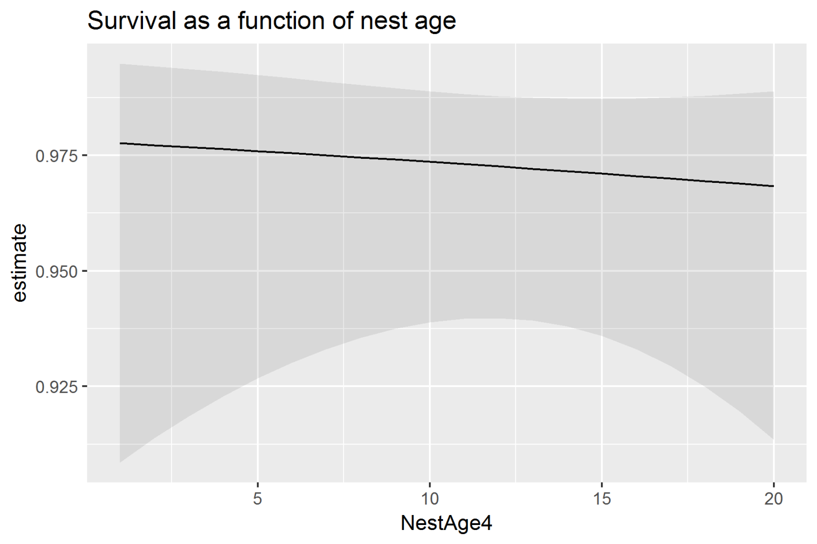
```
# First get the all.diff.index values for each day of the s
get.real(mod.res, param="S", se=TRUE)
# we see that all.diff.index==1 is for nest survival on day

# we will predict then the DSR for day 1 at various ages
pred.ages <- data.frame(NestAge1=1:20, index=1)
covariate.predictions(mod.res, data=pred.ages )$estimates[1
```

```
> # First get the all.diff.index values for each day of the
> get.real(mod.res, param="S", se=TRUE)
              all.diff.index par.index  estimate        se
S g1 a0 t1                 1         1 0.9813396 0.0222450 0
...
> # we will predict then the DSR for day 1 at various ages
```

Analysis of killdeer data with nest age II

```
> pred.ages <- data.frame(NestAge1=1:20, index=1)
> covariate.predictions(mod.res, data=pred.ages )$estimates
   vcv.index model.index par.index NestAge1 index  estimate
1          1           1         1        1     1 0.9776096
2          1           1         1        2     1 0.9771941
3          1           1         1        3     1 0.9767711
4          1           1         1        4     1 0.9763404
5          1           1         1        5     1 0.9759019
6          1           1         1        6     1 0.9754555
7          1           1         1        7     1 0.9750010
8          1           1         1        8     1 0.9745384
9          1           1         1        9     1 0.9740674
10         1           1         1       10     1 0.9735879
```

Analysis of killdeer data with nest age



Analysis of killdeer data with nest age

Use model averaging to investigate if nest age is a useful covariate.

```
> collect.models(type="Nest")
        model npar     AICc DeltaAICc    weight Deviance
1       S(~1)    1 44.52951  0.000000 0.7220459 42.51028
2 S(~NestAge)    2 46.43878  1.909265 0.2779541 42.38081
```

Exercise - Sherry - 3

Articles

Editor's Choice

Journal of Avian Biology 46: 559–569, 2015
doi: 10.1111/jav.00536

Subject Editor: Peter Arcese. Editor-in-Chief: Thomas Alerstam. Accepted 5 April 2015

Impacts of nest predators and weather on reproductive success and population limitation in a long-distance migratory songbird

Thomas W. Sherry, Scott Wilson, Sarah Hunter and Richard T. Holmes

Exercise - Sherry - 3

Refer t0 Table 1b.

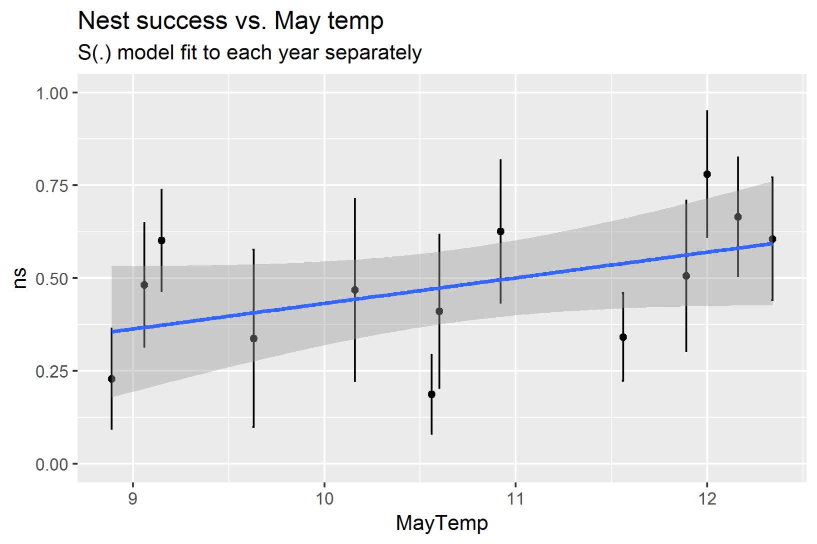
1. Reproduce Table 1b
2. Look at estimated beta from top model and compare to results in paper.

Exercise - Sherry - 4

Refer to Figure 2.

1. Reproduce Figure 2. Note that they analyzed each year separately with a simple $S \sim 1$ model.

Exercise - Sherry - 4



Summary I

- Apparent nest success is positively biased because of failure to account for exposure.
- Mayfield method is an approximate method that assumes constant DSR but is unable to account for covariates
- Modern modelling uses maximum likelihood estimation
  - Available in n *MARK*, *RMark*, and logisitic exposure models.
  - Tradeoff between flexibility and ease of use
- Goodness-of-fit is underdeveloped for nest success models, but see `http://www.montana.edu/rotella/nestsurv/`
- Random effects can be implemented in *MARK* and logistic exposure models and Bayesian methods