Design and Analysis of Nest Survival Studies
Part 2
Logistic Exposure Models

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

1. Basic logexp() models

2. Nest and Survey covariates using logexp() models

3. Nest Age covariates using logexp() models

4. Summary - logexp() models

Logistic exposure models in *R*

Logistic Exposure Models

- Uses only *R* to do analysis - *glm()* with special link function created by Mark Herzog and Ben Bolker after the *SAS* code by Terry Shaffer (2004).
- Need to load a function and preprocess the data.
- Very close approximation for time/age dependent models.
- Need to have a deep understanding of *R* to use effectively.
- Relatively easy to add random effects.

Logistic exposure models in *R* I

Extension of link function for *glm()* and related functions.
Refer to *logistic.exposure.R* file

```
logexp <- function(exposure = 1)
{
  linkfun <- function(mu) qlogis(mu^(1/exposure))
  ## FIXME: is there some trick we can play here to allow
  ##   evaluation in the context of the 'data' argument?
  linkinv <- function(eta)  plogis(eta)^exposure
  logit_mu_eta <- function(eta) {
    ifelse(abs(eta)>30,.Machine$double.eps,
           exp(eta)/(1+exp(eta))^2)
  }
  mu.eta <- function(eta) {
    exposure * plogis(eta)^(exposure-1) *
      logit_mu_eta(eta)
  }
  valideta <- function(eta) TRUE
  link <- paste("logexp(", deparse(substitute(exposure)), "
```

Logistic exposure models in *R* II

```
                     sep="")
  structure(list(linkfun = linkfun, linkinv = linkinv,
                      mu.eta = mu.eta, valideta = valideta,
                      name = link),
                 class = "link-glm")
}
```

Basic idea is that survival over x days is DSR^x.
PROBLEM. The exposure data is separate from the *data=xxx* argument in the *glm()* call which has implications for predictions in some cases.

Logistic exposure models in *R*

Basic steps in analysis:

- Input the nest data and preprocess the data (to get the proper effective sample size).
  - Expand to daily records when the nest is known to be "alive"
  - One final record when nest fails but date of failure is unknown.
  - Carefully define day and nest age.
- Define factors and covariates data frame
- Fit some model using the *glm()* function.
- Extract information from returned object in the usual way
- Model selection using the *AICmodavg* package.
- Obtain predictions on logit scale and transform to regular scale
- Plot/report the results

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.

Analysis of killdeer data using logistic exposure

Open the *killdeer-le.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 logistic exposure

Expand the data to get the effective sample size as noted in GIM Chapter 17, page 17-8..
Each day the nest is "alive" generates a single record.
Final interval when nest fails generates a single record.

```
# We expand the data to generate the effective sample size
killdata2 <- expand.nest.data(killdata)
killdata2
```

Analysis of killdeer data using logistic exposure I

Consider how the data are expanded:

Original data:

```
  id     FirstFound LastPresent LastChecked  Fate  Freq
1 /*A*/           1           9           9     0     1
```

Expanded data:

```
                  First     Last  Last             Exp
    id   Found  Present  Checked  Fate  Freq  Day  osure  Survive
 /*A*/       1        9        9     0     1    1      1        1
 /*A*/       1        9        9     0     1    2      1        1
 /*A*/       1        9        9     0     1    3      1        1
 /*A*/       1        9        9     0     1    4      1        1
 /*A*/       1        9        9     0     1    5      1        1
 /*A*/       1        9        9     0     1    6      1        1
```

Analysis of killdeer data using logistic exposure II

```
/*A*/    1    9    9    0    1    7    1    1
/*A*/    1    9    9    0    1    8    1    1
```

Consider how the data are expanded:

Original data:

```
  id     FirstFound LastPresent LastChecked  Fate  Freq
2 /*B*/           5           5           9     1     1
```

Expanded data:

```
                     First     Last   Last            Exp
         id   Found Present Checked Fate Freq Day osure Survive
  /*B*/      5       5        9    1    1    7     4     0
```

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 (!)

Analysis of killdeer data using logistic exposure I

Use *glm()* to analyze the data.

```
fit.Sdot <- glm(Survive~1,
          family=binomial(link=logexp(killdata2$Exposure)),
          data=killdata2)
summary(fit.Sdot)
```

Notice that you analyze the SURVIVAL and not the failure (as in *RMark*).
Notice how you specify the exposure separately from the *data*=.
This has (bad) implications if you subset the data.

Analysis of killdeer data using logistic exposure II

This gives.

```
> summary(fit.Sdot)
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   3.5570     0.4085   8.708   <2e-16 ***

    Null deviance: 54.491  on 209  degrees of freedom
Residual deviance: 42.510  on 209  degrees of freedom
AIC: 44.51

> -2*logLik(fit.Sdot)
'log Lik.' 42.51028 (df=1)
```

This is the DSR on the LOGIT scale.

Analysis of killdeer data using logistic exposure III

Compare with output from *RMark*

```
Npar :  1
## -2lnL:  42.51028
## AICc :  44.52951
##
## Beta
##                  estimate        se      lcl      ucl
## S:(Intercept) 3.557002 0.4141776 2.745214 4.368791
```

Analysis of killdeer data using logistic exposure I

We need to back transform the estimated DSR ourselves (groan).

```
# Convert the logit(DSR) to DSR
DSR <- expit(coef(fit.Sdot))
DSR.se <- arm::se.coef(fit.Sdot)*DSR*(1-DSR)
cat("DSR ", DSR, "(SE ", DSR.se, ")\n")

# Find confidence intervals by taking expit of confit of c
expit(confint(fit.Sdot))
```

This uses the delta method to find the se of a function of the parameters

Analysis of killdeer data using logistic exposure II

```
> arm::se.coef(fit.Sdot)
(Intercept)
  0.4084682
> DSR.se <- arm::se.coef(fit.Sdot)*DSR*(1-DSR)
> cat("DSR ", DSR, "(SE ", DSR.se, "\n")
DSR  0.9722669 (SE  0.01101394

> # Find confidence intervals by taking expit of confit of
> confint(fit.Sdot)
   2.5 %   97.5 %
2.835970 4.488518

> # Find confidence intervals by taking expit of confit of
> expit(confint(fit.Sdot))
    2.5 %    97.5 %
```

```
0.9445889 0.9888876
```

Compare with output from *RMark*

```
##                  estimate         se        lcl        ucl fixed
## S g1 a0 t1 0.9722669 0.0111679 0.9396425 0.9874919
```

CIs differ because of different method of computing ci of beta coefficients.

Analysis of killdeer data using logistic exposure I

We need to compute the nest survival ourselves (groan)

```
# Compute the nest survival
days <- 39
NS <- DSR^days
NS.se <- DSR.se * days * DSR^(days-1)
cat("NS ", days," days ", NS, "(SE ", NS.se, ")\n")
```

This gives.

```
> cat("NS ", days," days ", NS, "(SE ", NS.se, ")\n")
NS  39  days  0.3339133 (SE  0.1475216 )
```

Compare to *RMark*

```
> 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 logistic exposure I

Linear trend in DSR

```
fit.linear <- glm(Survive~Day,
           family=binomial(link=logexp(killdata2$Exposure)),
           data=killdata2)
summary(fit.linear)
```

This gives:

```
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  4.97938    1.35377   3.678 0.000235 ***
Day         -0.06149    0.05106  -1.204 0.228446

AIC: 44.997
```

Analysis of killdeer data using logistic exposure II

Compare to *RMark* values:

```
## Name : S(~Time)
##
## Npar :  2
## -2lnL:  40.97847
## AICc :  45.03644
##
## Beta
##                   estimate        se        lcl       ucl
## S:(Intercept)  4.9372980 1.3402857  2.3103379 7.5642581
## S:Time        -0.0622855 0.0526986 -0.1655748 0.0410037
```

Results differ slight because of treatment of date for final intervals.

Analysis of killdeer data using logistic exposure I

Linear trend in DSR - predicting DSR. We can use the predict function here because the default exposure is 1 day.

```
# We now want to predict the DSR for days 1..39
# In this case predict() will work on the logit scale, but
pred.data <- data.frame(Day=1:39)
logit.dsr.pred <- predict(fit.linear, newdata=pred.data, se

# put these together in a data frame
dsr <- cbind(pred.data, logit.dsr=logit.dsr.pred$fit, logit
head(dsr)
dsr$dsr <- expit(dsr$logit.dsr)
dsr$dsr.se <- dsr$logit.dsr.se* dsr$dsr * (1-dsr$dsr)
head(dsr)
```

Analysis of killdeer data using logistic exposure II

This gives:

```
> head(dsr)
  Day logit.dsr logit.dsr.se       dsr      dsr.se
1   1  4.917891     1.305189 0.9927386 0.009408717
2   2  4.856401     1.256804 0.9922816 0.009625636
3   3  4.794910     1.208639 0.9917961 0.009834179
4   4  4.733420     1.160720 0.9912804 0.010032802
5   5  4.671930     1.113081 0.9907325 0.010219886
6   6  4.610440     1.065758 0.9901505 0.010393754
```

Analysis of killdeer data using logistic exposure III

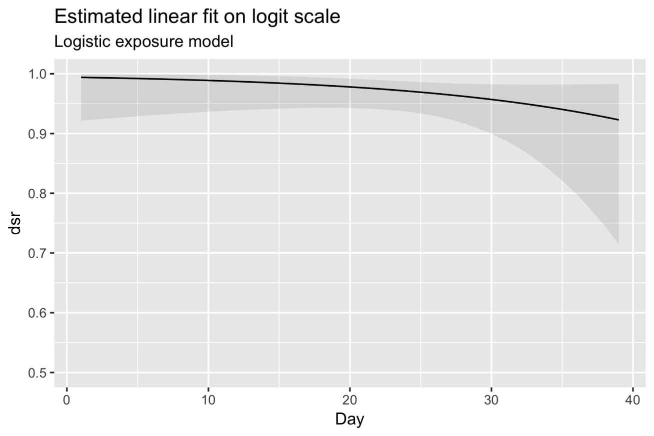
Compare to *RMark* values:

```
##                   estimate        se       lcl       ucl fi
## S g1 a0 t1   0.9928771 0.0094787 0.9097296 0.9994816
## S g1 a1 t2   0.9924229 0.0097027 0.9126190 0.9993915
## S g1 a2 t3   0.9919398 0.0099182 0.9153905 0.9992862
## S g1 a3 t4   0.9914263 0.0101238 0.9180448 0.9991630
## S g1 a4 t5   0.9908803 0.0103177 0.9205823 0.9990191
## S g1 a5 t6   0.9902999 0.0104980 0.9230028 0.9988512
```

Results differ slight because of treatment of date for final intervals.

Analysis of killdeer data using logistic exposure I

We can plot in the usual way.



Why is the fit curved?

Analysis of killdeer data using logistic exposure I

Quadratic trend in DSR - similar to linear trend.
We can use the predict function here because the default exposure is 1 day.
Consult code.

Analysis of killdeer data using logistic exposure I

S varies in first/second half of the study.

```
killdata2$studyhalf <- car::recode(killdata2$Day,
                                   " lo:20='1st'; 21:hi='2nd'")
head(killdata2)

fit.2part <- glm(Survive~studyhalf,
             family=binomial(link=logexp(killdata2$Exposure)),
             data=killdata2)
summary(fit.2part)
```

Analysis of killdeer data using logistic exposure II

This gives

```
              Estimate Std. Error z value Pr(>|z|)
(Intercept)     3.8604     0.7077   5.455 4.89e-08 ***
studyhalf2nd   -0.4986     0.8666  -0.575    0.565

AIC: 46.173
```

Analysis of killdeer data using logistic exposure III

Compare to *RMark* results

```
Npar :  2
## -2lnL:  41.92815
## AICc :  45.98612
##
## Beta
##                    estimate        se       lcl      ucl
## S:(Intercept)    3.9413016 0.7140400  2.541783 5.340820
## S:studyhalf2nd -0.6514382 0.8772159 -2.370781 1.067905
```

Ambiguity about interval that straddle the boundary between first and second half as the *Day* variable is the midpoint of the interval.

Analysis of killdeer data using logistic exposure I

The predictions are made in the usual way.

Analysis of killdeer data using logistic exposure

Model averaging – use *AICcmodavg* package (or similar packages)

Analysis of killdeer data using logistic exposure I

Open the *killdeer-modavg-le.r* script.

We will compare four models

Data read in the usual way.
Expand data in the usual way.
Add relevant variables to the data frame.

Analysis of killdeer data using logistic exposure II

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

# We expand the data to generate the effective sample size
killdata2 <- expand.nest.data(killdata)
head(killdata2)

# Add the Day2 term and the first/second half variables
killdata2$Day2 <- (killdata2$Day-20)^2
killdata2$studyhalf <- car::recode(killdata2$Day,
              " lo:20='1st'; 21:hi='2nd'")
```

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

```
model.list.csv <- textConnection(
" S
 ~1
 ~Day
 ~Day+Day2
 ~studyhalf
")

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

Analysis of killdeer data using logistic exposure I

Fit all of the models in the model set:

```
model.fits <- plyr::dlply(model.list, c("S","model.number")
                         function(x,input.data, input.ddl){
  cat("\n\n***** Starting ", unlist(x), "\n")

  fit <- glm(formula=as.formula(paste("Survive", eval(x$S))
        family=binomial(link=logexp(input.data$Exposure)),
        data=input.data)
  fit

},input.data=killdata2)
```

R successively fits the models and stores results in a big list

Analysis of killdeer data using logistic exposure I

Compute the AIC table.

```
AICcmodavg::aictab(model.fits)
```

```
Model selection based on AICc:

              K  AICc Delta_AICc AICcWt Cum.Wt     LL
~Day+Day2.3   3 43.88       0.00   0.37   0.37 -18.88
~1.1          1 44.53       0.65   0.27   0.64 -21.26
~Day.2        2 44.69       0.82   0.25   0.89 -20.32
~studyhalf.4 2 46.23       2.35   0.11   1.00 -21.09
```

Analysis of killdeer data using logistic exposure I

Get model average value of DSR using custom model averaging facilities.
Multi-step process

- Estimate the DSR for each model at each day
- Extract model information
- Use custom model averaging function.

Analysis of killdeer data using logistic exposure I

1. Estimate DSR for each model for each day

```
pred.data       <- data.frame(Day=1:39)
pred.data$Day2 <- (pred.data$Day-20)^2  # we need to match
pred.data$studyhalf <- car::recode(pred.data$Day,
                        " lo:20='1st'; 21:hi='2nd'")

dsr.indiv <- plyr::ldply(model.fits, function(x,pred.data)
   # get the predictions on the logit scale and then back
    logit.dsr.pred <- predict(x, newdata=pred.data,
                        se.fit=TRUE)

    # put these together in a data frame
    dsr <- cbind(pred.data, logit.dsr=logit.dsr.pred$fit,
                      logit.dsr.se=logit.dsr.pred$se.fit)
    dsr$dsr    <- expit(dsr$logit.dsr)
    dsr$dsr.se <- dsr$logit.dsr.se* dsr$dsr * (1-dsr$dsr)
    dsr
},pred.data=pred.data)
```

Analysis of killdeer data using logistic exposure II

```
> head(plotdata.indiv)
   S model.number Day Day2 studyhalf logit.dsr logit.dsr.se
1 ~1            1   1  361       1st  3.557002     0.408469
2 ~1            1   2  324       1st  3.557002     0.408469
3 ~1            1   3  289       1st  3.557002     0.408469
4 ~1            1   4  256       1st  3.557002     0.408469
5 ~1            1   5  225       1st  3.557002     0.408469
6 ~1            1   6  196       1st  3.557002     0.408469
```

2. Extract model information (log likelihood, number of parameters, sample size)

```
model.info <- plyr::ldply(model.fits, function(x){
     #browser()
     logL <- logLik(x)
     K    <- length(coef(x))
     nobs <- nrow(x$data)
     data.frame(logL=logL, K=K, nobs=nobs)
})
model.info
```

```
> model.info
            S model.number      logL K nobs
1          ~1            1 -21.25514 1  210
2        ~Day            2 -20.31785 2  210
3  ~Day+Day2            3 -18.88091 3  210
4 ~studyhalf            4 -21.08636 2  210
```

Analysis of killdeer data using logistic exposure I

3. Use Custom model averaging function for each day

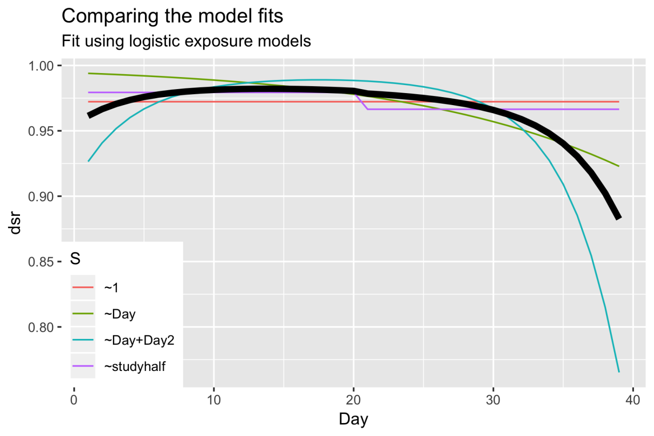
```
dsr.ma <- plyr::ddply(dsr.indiv, c("Day"), function(x, mode
   # merge the model information with the estimates
   x <-merge(x, model.info)
   # get the model averaged values
   #browser()
   ma <- AICcmodavg::modavgCustom(x$logL, x$K,
                    modnames=x$S,
                    nobs=x$nobs,
                    estimate=x$dsr, se=x$dsr.se)
   data.frame(dsr=ma$Mod.avg.est,
         dsr.se=ma$Uncond.SE,
         dsr.lcl=ma$Lower.CL,
         dsr.ucl=ma$Upper.CL)
},model.info=model.info)
```

The individual DSR values are model averaged:

Analysis of killdeer data using logistic exposure II

```
> head(dsr.ma)
  Day       dsr     dsr.se   dsr.lcl  dsr.ucl
1   1 0.9614451 0.06878004 0.8266387 1.096251
2   2 0.9666242 0.05174868 0.8651987 1.068050
3   3 0.9705807 0.03934304 0.8934697 1.047692
4   4 0.9736046 0.03041289 0.9139964 1.033213
5   5 0.9759166 0.02408579 0.9287093 1.023124
6   6 0.9776827 0.01970441 0.9390627 1.016303
>
```

Plotting the individual curves and the model averaged curve.



It is also possible to model average the derived parameters such as nest success in a similar way (not shown)

Exercise - Sherry - 1 - logistic exposure model

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 - logistic exposure model

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
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 easier to implement with logistic exposure models compared to *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. $Survive \sim Distance$).
- Categorical covariates
  - Enter as an alphanumeric columns in the nest data frame and declare as a factor.
  - Specify variable name in formula, e.g. $Survive \sim Treatment$).

Nest level categorical covariates - logexp() models I

Read in the **mallard** dataset.

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

mallard2 <- expand.nest.data(malldata)

malldata2$Habitat <- factor(malldata2$Habitat)
```

Not necessary that categorical variables be declared as factors

Nest level categorical covariates -logexp() models I

Use the categorical variable in the model

```
mod.hab <-  glm(Survive~Habitat,
           family=binomial(link=logexp(malldata2$Exposure)),
           data=malldata2)
summary(mod.hab)
```

Nest level categorical covariates -logexp() models II

```
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.86293    0.09508  30.111   <2e-16 ***
HabitatP     0.22268    0.12234   1.820   0.0687 .
HabitatR     0.21111    0.22719   0.929   0.3528
HabitatW     0.09291    0.23509   0.395   0.6927

 Residual deviance: 1564.0  on 6122  degrees of freedom
AIC: 1572
```

Nest level categorical covariates -logexp() models III

Compare to *RMark* output:

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

Npar :  4
-2lnL:  1563.951
AICc :  1571.957

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
```

Nest level categorical covariates -logexp() models I

DO NOT TRUST the output from the summary table as it depends on the (hidden) contrast matrix used to set up the indicator variables.
Values can change depending on user's configuration without warning.

Two options:

- Make predictions as before for simple models
- Use the *emmeans* package (with some modification for the custom link function) as the results are independent of the (hidden) contrast matrix or reference level used.

The latter is preferred when need to average levels of other factors (i.e. marginal estimates)

Nest level categorical covariates -logexp() models I

Using the *predict()*

```
# You can make predictions just as before
pred.data <- data.frame(Habitat=unique(malldata2$Habitat))
logit.dsr.pred.hab <- predict(mod.hab,
              newdata=pred.data, se.fit=TRUE)

# put these together in a data frame
dsr.hab <- cbind(pred.data,
              logit.dsr=logit.dsr.pred.hab$fit,
              logit.dsr.se=logit.dsr.pred.hab$se.fit)
dsr.hab$dsr <- expit(dsr.hab$logit.dsr)
dsr.hab$dsr.se <- dsr.hab$logit.dsr.se* dsr.hab$dsr *
             (1-dsr.hab$dsr)
dsr.hab
```

Nest level categorical covariates -logexp() models II

This gives

```
> dsr.hab
  Habitat logit.dsr logit.dsr.se       dsr      dsr.se
1       R  3.074044   0.20633908 0.9558093 0.008715326
2       P  3.085609   0.07698050 0.9562952 0.003217376
3       N  2.862929   0.09507968 0.9459832 0.004858478
4       W  2.955842   0.21500819 0.9505389 0.010108552
```

Nest level categorical covariates -logexp() models III

RMark output

```
##                    estimate        se       lcl       ucl fixed
## S gN a0 t1 0.9459833 0.0050725 0.9351340 0.9551051
## S gP a0 t1 0.9562953 0.0033341 0.9492739 0.9623833
## S gR a0 t1 0.9558094 0.0090265 0.9343297 0.9704853
## S gW a0 t1 0.9505390 0.0105538 0.9252465 0.9675738
```

Using the *emmeans* package

- Set up the reference grid (the emmeans object)
- Get the CLD, or Pairwise differences on logit scale
- Get the CLD or pairwise effects on regular scale after regridding for custom link function

Nest level categorical covariates -logexp() models II

```
# extract the logit(DSR) for each habitat using emmeans
mod.hab.emmo <- emmeans::emmeans(mod.hab, ~Habitat)
dsr.logit <- CLD(mod.hab.emmo)
dsr.logit

# Compute the multiple comparison on the ordinary scale
# We need to update the transformation so we can get answe
mod.hab.rg <- update(ref_grid(mod.hab, at=list(exposure=1))
           tran = logexp())

mod.hab.emmo2 <- emmeans::emmeans(mod.hab.rg, ~Habitat)
CLD(mod.hab.emmo2)
dsr <- CLD(mod.hab.emmo2, type="response") # on DSR scale
dsr
```

Nest level categorical covariates -logexp() models III

```
On logit scale:
 Habitat emmean     SE  df asymp.LCL asymp.UCL .group
 N         2.86 0.0951 Inf      2.68      3.05  1
 W         2.96 0.2150 Inf      2.53      3.38  1
 R         3.07 0.2063 Inf      2.67      3.48  1
 P         3.09 0.0770 Inf      2.93      3.24  1

Results are given on the logexp(1) (not the response) scale
Confidence level used: 0.95
P value adjustment: tukey method for comparing a family of
significance level used: alpha = 0.05
```

Nest level categorical covariates -logexp() models IV

```
> dsr <- CLD(mod.hab.emmo2, type="response") # on DSR scale
> dsr
 Habitat  prob      SE  df asymp.LCL asymp.UCL .group
 N       0.946 0.00486 Inf     0.936     0.955  1
 W       0.951 0.01011 Inf     0.927     0.967  1
 R       0.956 0.00872 Inf     0.935     0.970  1
 P       0.956 0.00322 Inf     0.950     0.962  1
```

Nest level categorical covariates -logexp() models V

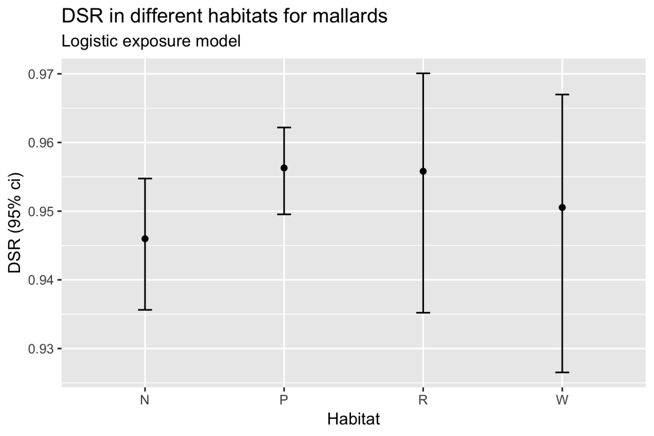
Compared to *RMark* output:

```
##                  estimate        se       lcl       ucl fixed
## S gN a0 t1 0.9459833 0.0050725 0.9351340 0.9551051
## S gP a0 t1 0.9562953 0.0033341 0.9492739 0.9623833
## S gR a0 t1 0.9558094 0.0090265 0.9343297 0.9704853
## S gW a0 t1 0.9505390 0.0105538 0.9252465 0.9675738
```

Former ci are wider to account for multiple comparisons.

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 <-  glm(Survive~1,
          family=binomial(link=logexp(malldata2$Exposure)),
          data=malldata2)
summary(mod.null)

AICcmodavg::aictab( list(mod.hab, mod.null))
```

This gives

```
     K    AICc Delta_AICc AICcWt Cum.Wt      LL
Mod2 1 1569.12       0.00   0.81   0.81 -783.56
Mod1 4 1571.96       2.84   0.19   1.00 -781.98
```

RMark gives:

```
> 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.

Example, effect of cover (Robel height) on DSR

```
mod.rob <- glm(Survive~Robel,
          family=binomial(link=logexp(malldata2$Exposure)),
          data=malldata2)summary(mod.rob)
```

Nest level continuous covariates II

```
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.90884    0.16827  17.287   <2e-16 ***
Robel        0.02727    0.04499   0.606    0.544
-
AIC: 1570.8
```

Estimated slope (on logit scale) is .027 (SE .047) and 95% ci for slope includes zero.

Nest level continuous covariates III

RMark output:

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

Npar :  2
-2lnL:  1566.773
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
```

Nest level continuous covariates I

We can get predictions of DSR at different Robel heights in the usual way.

```
pred.data <- data.frame(Robel=seq(min(malldata2$Robel),
         max(malldata2$Robel), length.out=50))
logit.dsr.pred.rob <- predict(mod.rob, newdata=pred.data,
          se.fit=TRUE)

# put these together in a data frame
dsr.rob <- cbind(pred.data,
          logit.dsr=logit.dsr.pred.rob$fit,
          logit.dsr.se=logit.dsr.pred.rob$se.fit)
dsr.rob$dsr <- expit(dsr.rob$logit.dsr)
dsr.rob$dsr.se <- dsr.rob$logit.dsr.se* dsr.rob$dsr *
          (1-dsr.rob$dsr)
dsr.rob
```

Nest level continuous covariates II

This gives

```
      Robel logit.dsr logit.dsr.se       dsr      dsr.se
1 0.6250000  2.925881   0.14202616 0.9491111 0.006859751
2 0.8010204  2.930681   0.13477235 0.9493425 0.006481385
3 0.9770408  2.935482   0.12759770 0.9495728 0.006109926
4 1.1530612  2.940282   0.12051636 0.9498022 0.005745981
5 1.3290816  2.945082   0.11354576 0.9500305 0.005390305
6 1.5051020  2.949882   0.10670764 0.9502579 0.005043838
```

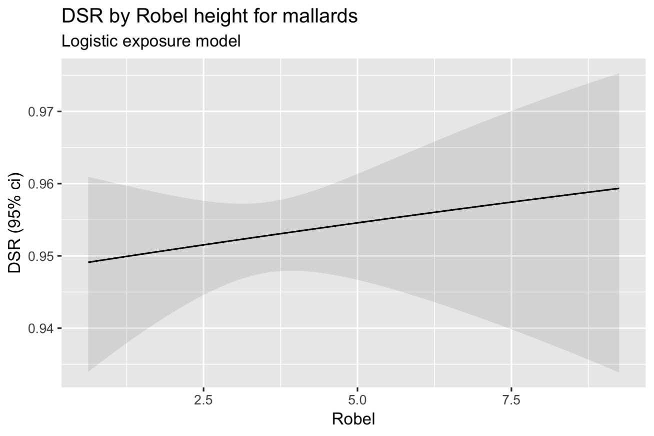
Nest level continuous covariates III

RMark gives:

```
 Robel index  estimate          se       lcl       ucl fixe
0.6250000     1 0.9491111 0.007112088 0.9332221 0.9613761
0.8010204     1 0.9493425 0.006720197 0.9344286 0.9610059
0.9770408     1 0.9495729 0.006335469 0.9356051 0.9606383
1.1530612     1 0.9498022 0.005958547 0.9367506 0.9602748
1.3290816     1 0.9500306 0.005590208 0.9378640 0.9599167
1.5051020     1 0.9502580 0.005231429 0.9389436 0.9595660
```

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.

Nest level continuous covariates I

Fit a null model and do AIC in the usual way.

```
     K    AICc Delta_AICc AICcWt Cum.Wt      LL
Mod2 1 1569.12       0.00    0.7    0.7 -783.56
Mod1 2 1570.77       1.66    0.3    1.0 -783.39
```

RMark gives

```
 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 data frame in the usual way for each nest and proceed similarly.

Exercise - Sherry - 2 - logexp() models

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

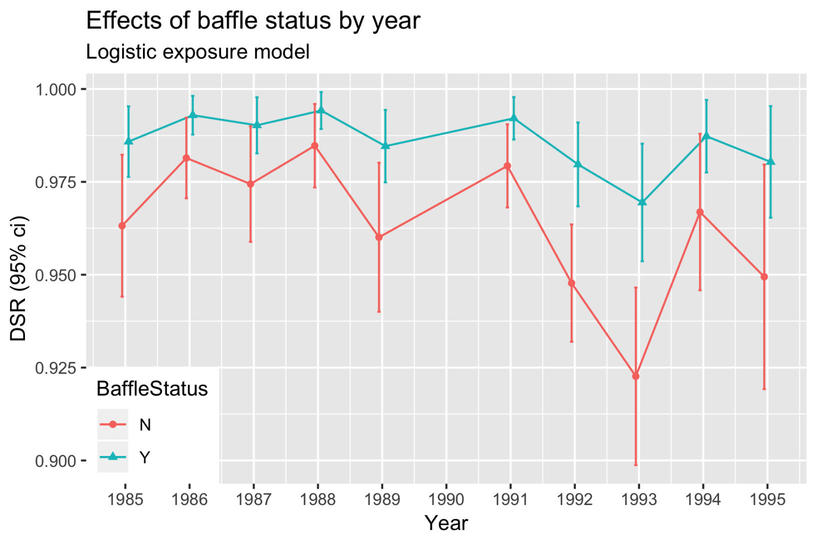
Subject Editor: Peter Arcese. Editor-in-Chief: Thomas Alerstam. Accepted 5 April 2015

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

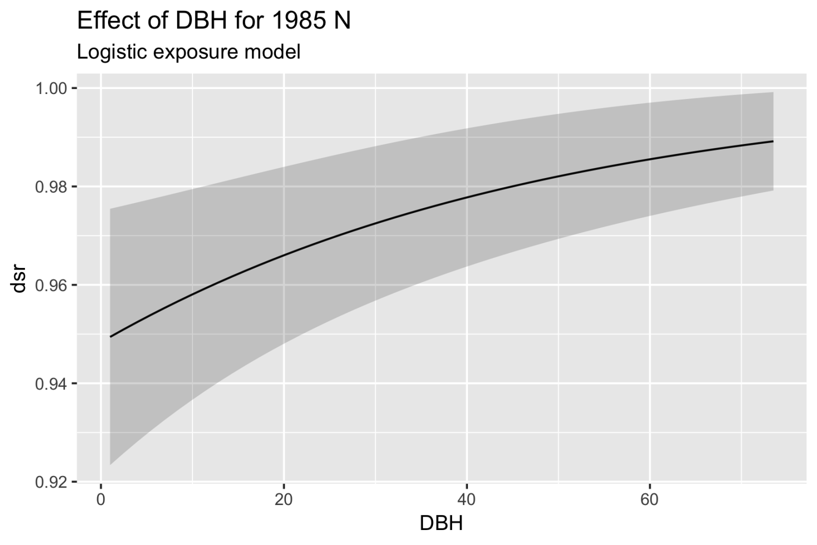
Thomas W. Sherry, Scott Wilson, Sarah Hunter and Richard T. Holmes

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

Exercise - Sherry - 2 - logexp() models



Exercise - Sherry - 2 - logexp() models



Nest x Time covariates - logexp() models

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

Much simpler in the logistic exposure models because you have a record for each day in the study for each nest (after expansion).

Some approximation is done for last interval where failure of a nest occurs but the time of the failure is unknown. The midpoint of the interval is used for the time of the study and nest age.

Age effects - logexp() models

The variable **AgeDay1** for the age of the nest on the first day of the season. The variable **AgeDay1** in the datafile.is then used to generate a variable **NestAge** for every day for every nest when the data is expanded. in the modelling.

Including age effects - logexp() models

Open the *killdeer.xlsx* workbook.
Open the *killdeer-age.R* script.

We read in the raw data and expand the 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)

killdata2 <- expand.nest.data(killdata)
```

Including age effects - logexp() models I

Look at expanded data and ages

```
      First
   id Found LastPresent LastChecked AgeDay1 Day NestAge
/*A*/     1           9           9       0   1       0
/*A*/     1           9           9       0   2       1
/*A*/     1           9           9       0   3       2
/*A*/     1           9           9       0   4       3
/*A*/     1           9           9       0   5       4
/*A*/     1           9           9       0   6       5
/*A*/     1           9           9       0   7       6
/*A*/     1           9           9       0   8       7

/*B*/     5           5           9      -2   7       4

/*C*/     5          40          40      -3   5       1
```

Including age effects - logexp() models II

```
/*C*/    5    40    40    -3   6    2
/*C*/    5    40    40    -3   7    3
/*C*/    5    40    40    -3   8    4
/*C*/    5    40    40    -3   9    5
```

Note that age assigned to midpoint of last interval.

Including age effects - logexp() models I

Fit a model with *DSR* as a function of nest age

```
mod.age <-  glm(Survive~NestAge,
           family=binomial(link=logexp(killdata2$Exposure)),
           data=killdata2)
summary(mod.age)
```

Including age effects - logexp() models II

This gives the output

```
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.89120    0.79883   4.871 1.11e-06 ***
NestAge     -0.02589    0.04996  -0.518    0.604

-AIC: 46.257
```

Including age effects - logexp() models III

RMark output:

```
> 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
```

We predict the relationship between DSR and nest age in the usual way.

```
# predict surival as a function of nest age
pred.data <- data.frame(NestAge=1:10)
logit.dsr.pred.age <- predict(mod.age, newdata=pred.data, s

# put these together in a data frame
dsr.age <- cbind(pred.data, logit.dsr=logit.dsr.pred.age$fi
head(dsr.age)
dsr.age$dsr <- expit(dsr.age$logit.dsr)
dsr.age$dsr.se <- dsr.age$logit.dsr.se* dsr.age$dsr * (1-ds
head(dsr.age)
```

Analysis of killdeer data with nest age II

```
> head(dsr.age)
  NestAge logit.dsr logit.dsr.se       dsr     dsr.se
1       1  3.865309    0.7563272 0.9794737 0.01520592
2       2  3.839419    0.7147924 0.9789467 0.01473193
3       3  3.813528    0.6744015 0.9784064 0.01424829
4       4  3.787638    0.6353727 0.9778526 0.01376021
5       5  3.761747    0.5979726 0.9772849 0.01327448
6       6  3.735857    0.5625264 0.9767030 0.01279988
```

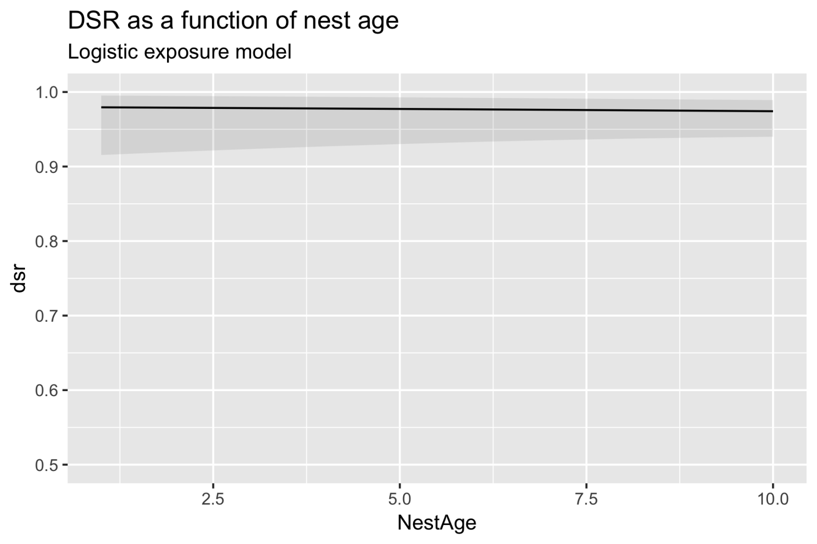
Analysis of killdeer data with nest age III

RMark output

```
NestAge1 index  estimate          se        lcl        ucl fix
 1      1 0.9776096 0.01653021 0.9085721 0.9948142
 2      1 0.9771941 0.01588040 0.9138054 0.9942588
3      1 0.9767711 0.01523016 0.9185801 0.9936599
4      1 0.9763404 0.01458597 0.9229012 0.9930196
5      1 0.9759019 0.01395614 0.9267692 0.9923424
6      1 0.9754555 0.01335112 0.9301786 0.9916358
```

Differences due to last interval.

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.

```
      K  AICc Delta_AICc AICcWt Cum.Wt     LL
Mod2 1 44.53       0.00   0.71   0.71 -21.26
Mod1 2 46.31       1.79   0.29   1.00 -21.13
```

RMark output

```
> 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 - logexp() model

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 - logexp() model

Refer t0 Table 1b.

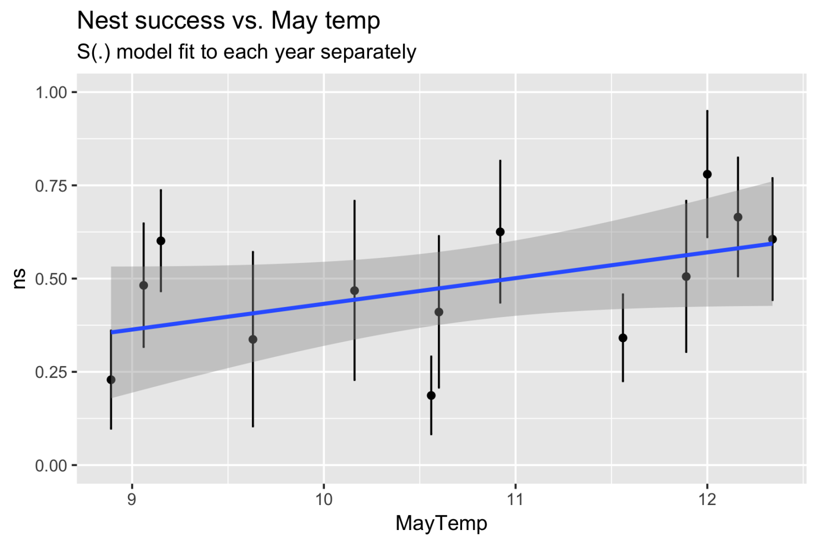
1. Reproduce Table 1b
2. Look at estimated beta from top model and compare to results in paper.

Exercise - Sherry - 4 - logexp() model

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 - logexp() model



Summary - logexp() models I

- Uses basis *R* functions with custom link function.
- Some slight differences in results from *MARK* or *RMark* when using models with *NestAge* or *Date* because last interval is assigned middle time/age in model rather than individual ages/dates for each day in the interval.
- Need more experience with *R* and understanding of basic *R* functions.
- Goodness-of-fit is underdeveloped for nest success models, but see `http://www.montana.edu/rotella/nestsurv/`
- Random effects can be (easily) implemented for logistic exposure models (and Bayesian methods (contact me)