Statistics with Sparrows - many models, matrices, and some magic

Julia Schroeder and David Orme

2017

Day 4 Binomial errors

We introduced generalised linear models (GLM) using count data and the Poisson error structure. We're now going to look at using a different error structure that is important for fi ng propor onal data or binary data. One important thing to note is makes use of propor on data where we have the number of successes out of a number of trials (binomial data). This tells us a lot about the variance of the expected propor on and the weight we assign to different data points.

Endemicity on the Galapagos Islands

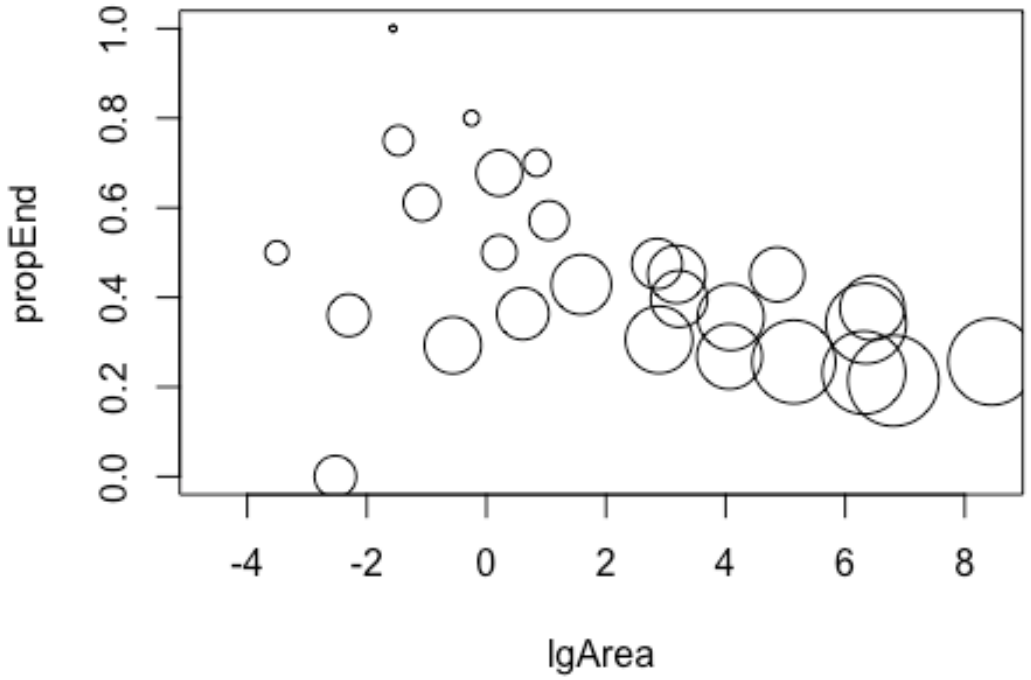
We'll introduce the model using a simple reanalaysis of the species richness data from the Galapagos. As well as the number of plant species, we also have a record of the number of endemic species on each island. This gives a binomial es mate of the propor on of endemics on each island. The ques on is – does the propor on of endemic species vary with island area?

```
rm(list=ls())
setwd("~/Box Sync/Teaching/MagicStats")

gala <- read.delim('gala.txt')
str(gala)
```

```
## 'data.frame':    30 obs. of  7 variables:
##  $ Species  : int  58 31 3 25 2 18 24 10 8 2 ...
##  $ Endemics : int  23 21 3 9 1 11 0 7 4 2 ...
##  $ Area     : num  25.09 1.24 0.21 0.1 0.05 ...
##  $ Elevation: int  346 109 114 46 77 119 93 168 71 112 ...
##  $ Nearest  : num  0.6 0.6 2.8 1.9 1.9 8 6 34.1 0.4 2.6 ...
##  $ Scruz    : num  0.6 26.3 58.7 47.4 1.9 ...
##  $ Adjacent : num  1.84 572.33 0.78 0.18 903.82 ...
```

We devided the number of endemic species by the log of the area to get a new variable that tells us about the endemic species richness.

```
gala$propEnd <- gala$Endemic / gala$Species
gala$lgArea <- log(gala$Area)
plot(propEnd ~ lgArea , data=gala, cex=log(Species/2))
```



The figure above uses the plot character size (controlled using cex for character expansion) to show the number of species behind each es mated propor on. It looks like you get fewer endemics on larger islands, but is this significant? For binomial data, we need to change the error family and link func on to *family=binomial(link=logit)*. Again, the logit link is the default, so we could omit it and just use family=binomial. For binomial models, we also have to let the model know what the number of species (the binomial denominator) are as well as the propor on endemic. There are two ways of doing this in R. One is to create a response variable as a matrix with two columns showing the number of successes (endemic species) and number of failures (in this case, non-endemic species) so that the row sums give the total number of species.

```
resp <- with(gala, cbind(Endemics, Species-Endemics))

galaMod <- glm(resp ~ lgArea, data=gala, family=binomial(link=logit))
```

Another way of doing this, which I think is easier, is to give the proportion on as the response variable and the total number of species using the weights option.

```
galaMod <- glm(propEnd ~ lgArea, weights=Species,
          data=gala, family=binomial(link=logit) )
```

As all other models, the binomial GLM has an analysis of deviance table and uses a χ^2 test

on the change in deviance rather than an F test. We haven't really done this a lot yet but it's good to know it's there. It tells us how much deviance is explained by lgArea, and how much residual error (or deviance) is left.

```
anova(galaMod, test='Chisq')
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: propEnd
##
## Terms added sequentially (first to last)
##
##
##        Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                      29     154.39
## lgArea  1   44.053        28     110.33 3.195e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
1-pchisq(44.053,1)
```

```
## [1] 3.196032e-11
```

And again we can look at the model coefficients and deviance explained.

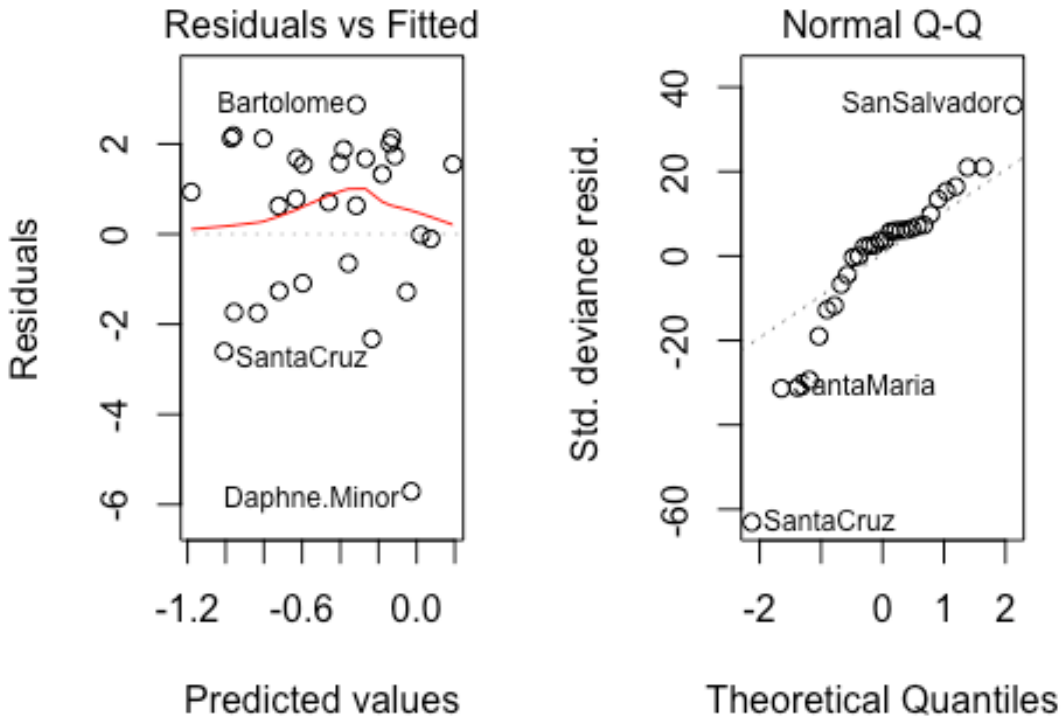
```
summary(galaMod)
```

```
##
## Call:
## glm(formula = propEnd ~ lgArea, family = binomial(link = logit),
##     data = gala, weights = Species)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.7089  -0.9793   0.8584   1.7203   2.8703
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.29358    0.08835  -3.323  0.00089 ***
## lgArea      -0.10494    0.01578  -6.652 2.89e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 154.39  on 29  degrees of freedom
## Residual deviance: 110.33  on 28  degrees of freedom
```

```
## AIC: 222.98
##
## Number of Fisher Scoring iterations: 4

(galaMod$null.deviance - galaMod$deviance)/galaMod$null.deviance

## [1] 0.2853442
```

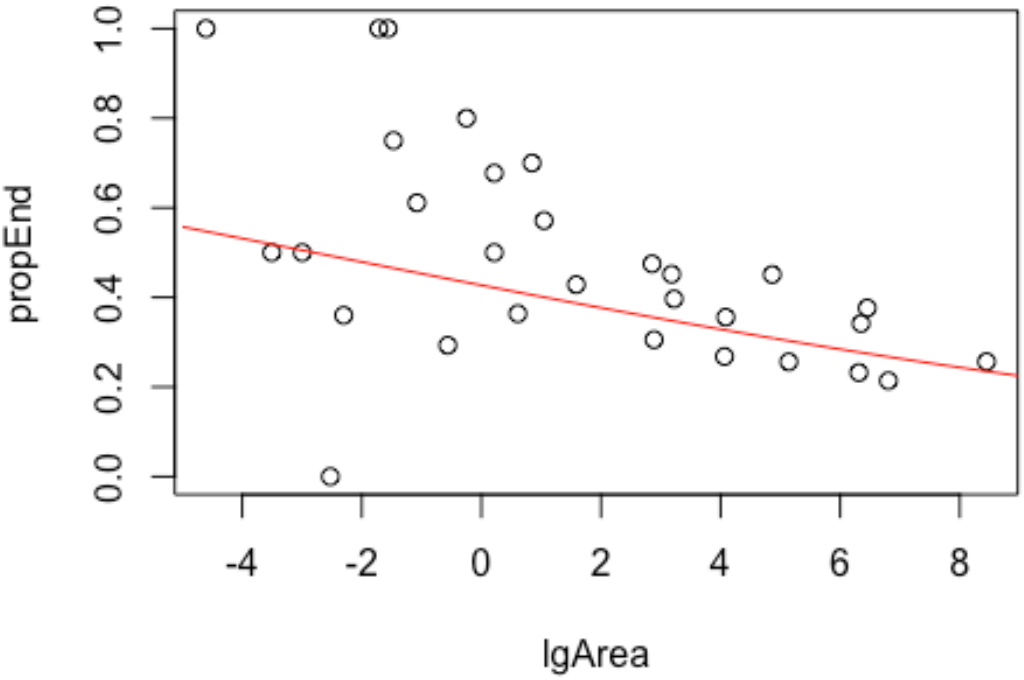
The diagnostic plots for this model aren't wonderful -- we won't worry about this now.

```
par(mfrow=c(1,2))
plot(galaMod, which=c(1,2))
```



Like for the poisson data in the last practical, we can work out the predicted values on the proportion scale the easy way:

```
# predict for a neat sequence of log area values
pred <- expand.grid(lgArea = seq(-5, 9, by=0.1))
pred$fit <- predict(galaMod, newdata=pred, type='response')

# plot the logged data and the model lines
plot(propEnd ~ lgArea, data=gala)
lines(fit ~ lgArea, data=pred, col='red')
```



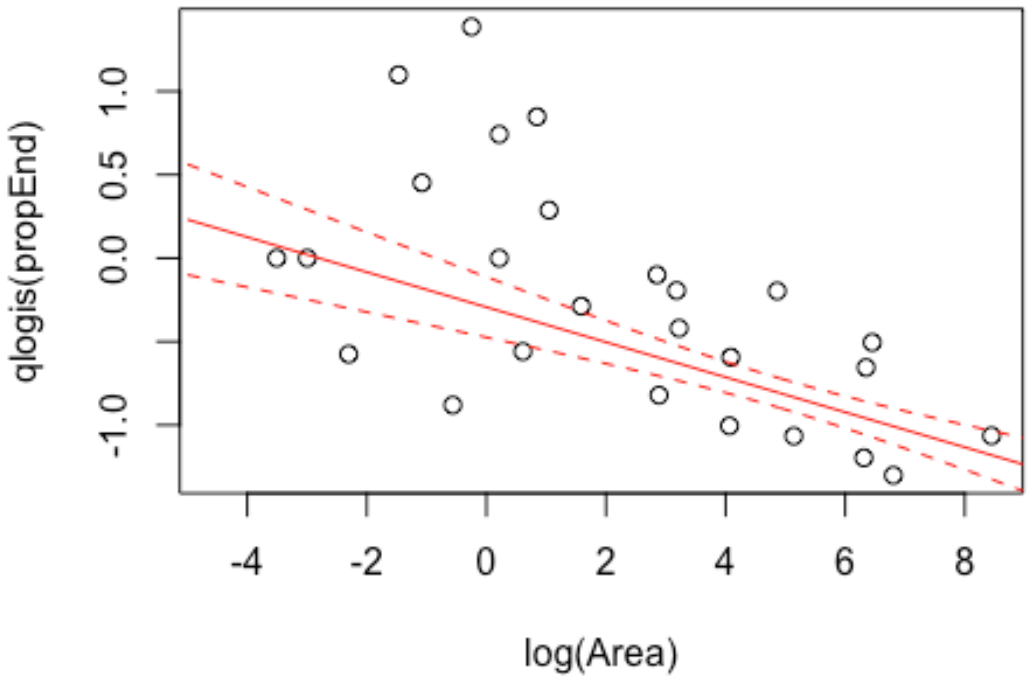
To get confidence limits, we need to use the inverse link function again. For the logit link, the two functions are:

$$\text{logit link} = \log\left(\frac{p}{1-p}\right), \qquad \text{logit link inverse} = \frac{e^x}{1+e^x}$$

There are handy functions to do these conversions: converts proprotions to logit values and converts logit predictions back to proportions.

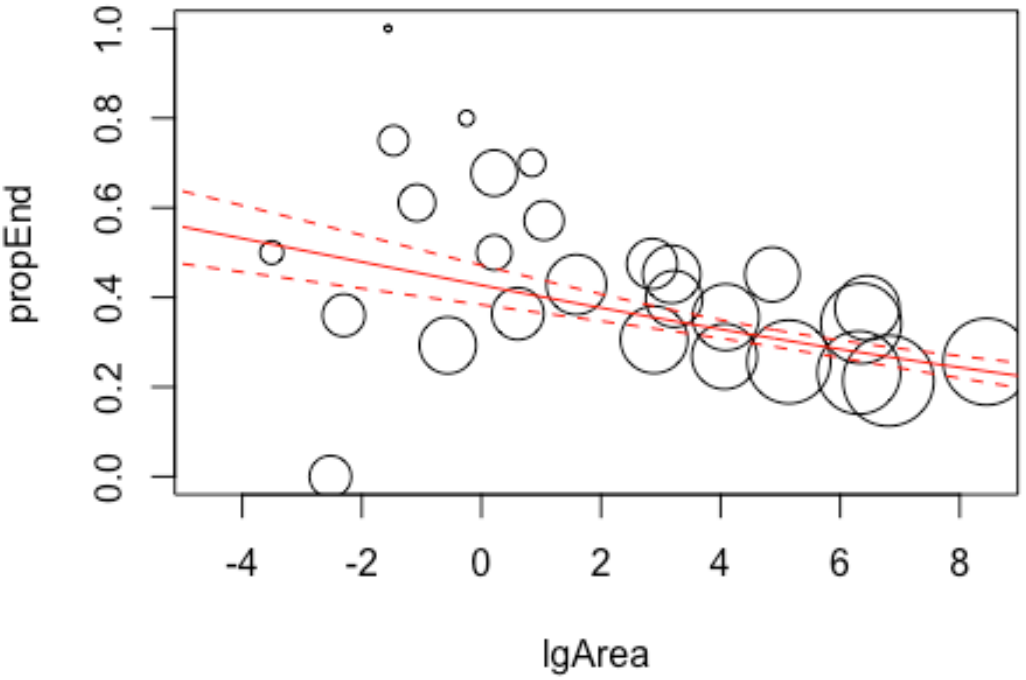
```
# predict for a neat sequence of log area values
pred <- expand.grid(lgArea = seq(-5, 9, by=0.1))
predMod <- predict(galaMod, newdata=pred, se.fit=TRUE)

# get the fit and confidence limits
pred$fit <- predMod$fit
pred$se.fit <- predMod$sefit
pred$confint <- predMod$se.fit * qt(0.975, df=galaMod$df.residual)

# plot the logit transformed proportion data and the model lines
plot(qlogis(propEnd) ~ log(Area), data=gala)
lines(fit ~ lgArea, data=pred, col='red')
```

```r
lines(fit + confint  ~ lgArea, data=pred, col='red', lty=2)
lines(fit - confint  ~ lgArea, data=pred, col='red', lty=2)
```



We can now back transform them on to the data. This means we need to know the inverse link function for our model, which for is .

```r
# plot the proportion data
plot(propEnd ~ lgArea , data=gala, cex=log(Species/2))
# add the link inverse transformed lines
lines(plogis(fit) ~ lgArea, data=pred, col='red')
lines(plogis(fit + confint)  ~ lgArea, data=pred, col='red', lty=2)
lines(plogis(fit - confint)  ~ lgArea, data=pred, col='red', lty=2)
```



Predicting threat in Galliformes

```
galliformes <- read.table('galliformesData.txt', header=TRUE)
str(galliformes)

## 'data.frame':    268 obs. of  8 variables:
##  $ Family   : Factor w/ 6 levels "Cracidae","Megapodiidae",..: 1 2 2 5 4 4
5 5 5 5 ...
##  $ CName    : Factor w/ 268 levels "Aceh_Pheasant",..: 251 249 41 71 258
19 10 66 193 203 ...
##  $ SName    : Factor w/ 268 levels "Aburria_aburri",..: 1 2 3 4 5 6 7 8 9
10 ...
##  $ Status04 : Factor w/ 5 levels "1_(LC)","2_(NT)",..: 2 1 3 3 3 1 1 1 1 1
...
##  $ Range    : int  778240 205731 2571 202631 315160 658866 1542970
11577357 446540 345484 ...
##  $ Mass     : num  1423 1600 1600 1418 815 ...
##  $ Clutch   : num  3.25 20 NA 2.5 12 NA 11.3 9.5 11 12.3 ...
##  $ ElevRange: int  2000 2050 1000 1200 NA NA 3300 4500 2400 2700 ...

galliformes <- na.omit(galliformes)
```

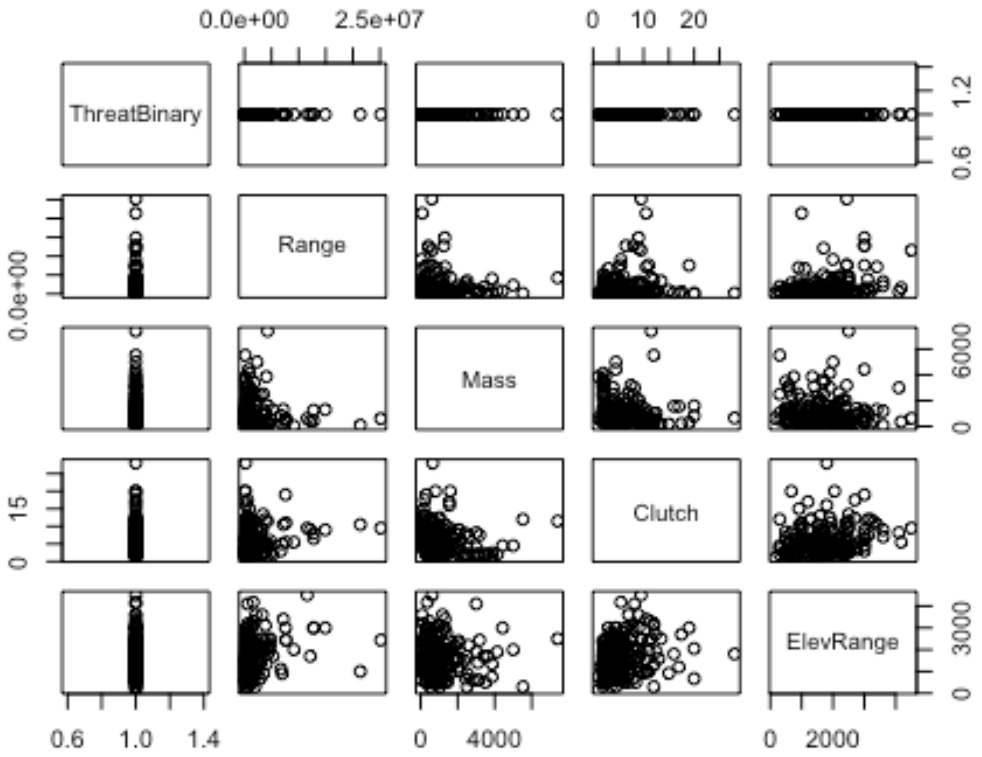
The first thing to do is to convert the threat status column from IUCN categories to a simple threatened (1) or not threatened (0) numeric variable.

```
galliformes$ThreatBinary <- ifelse(galliformes$Status04 %in% c("1 (LC)", "2
(NT)"), 0, 1)
```

Can you work out what this line of code does?

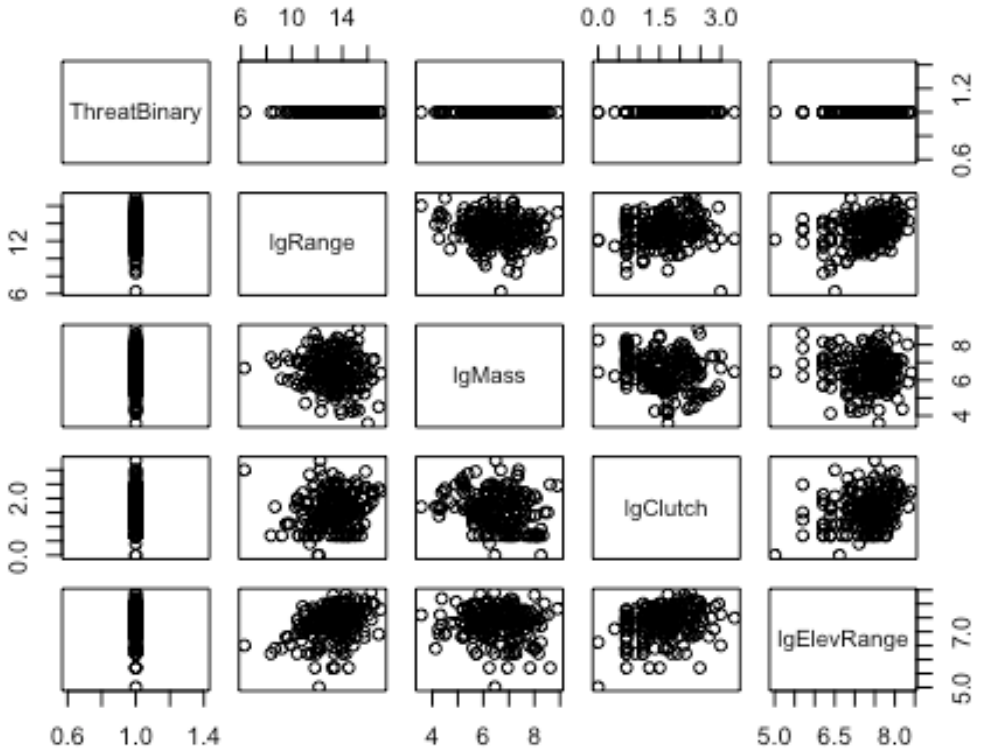
The life history variables we'll use are body mass, geographic range, clutch size and elevational range. If we check those, they all show strong right skew -- the points are all clumped over to the left:

```
pairs(ThreatBinary ~ Range + Mass + Clutch + ElevRange, data=galliformes)
```



So, log transformation to the rescue once again.

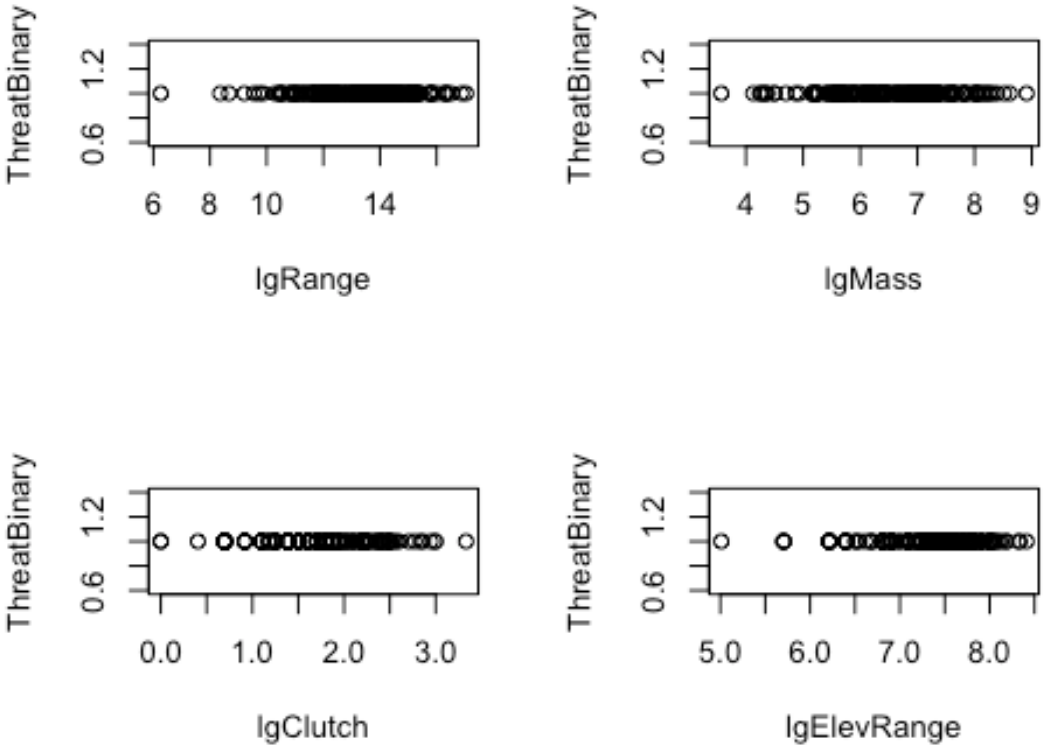
```
galliformes$lgMass <-  log(galliformes$Mass)
galliformes$lgRange <-  log(galliformes$Range)
galliformes$lgClutch <-  log(galliformes$Clutch)
galliformes$lgElevRange <-  log(galliformes$ElevRange)

pairs(ThreatBinary ~ lgRange +lgMass + lgClutch + lgElevRange,
data=galliformes)
```



Now we can look at some plots of the relationships to be modelled. Binary plots are hard to read but you can see that for most of those variables, the ones and zeros are concentrated

in different places along the x axis: as the variables change, the probability of being threatened changes.

```
par(mfrow=c(2,2))
plot(ThreatBinary ~ lgRange, data=galliformes)
plot(ThreatBinary ~ lgMass, data=galliformes)
plot(ThreatBinary ~ lgClutch, data=galliformes)
plot(ThreatBinary ~ lgElevRange, data=galliformes)
```



Now the hard bit -- and this is a trickier example: try and find the best model to explain threat status, starting with the model below, including all four variables and all two-way interactions. Try to use and ! An important point to note is that -- for -- it isn't necessary to provide weights to the binomial glm. Each data point represents a single case and has the same weight.

```
galliMod <- glm(ThreatBinary ~ (lgRange +lgMass + lgClutch + lgElevRange)^2,
                data=galliformes, family=binomial(link=logit))

## Warning: glm.fit: algorithm did not converge
```