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

Julia Schroeder, David Orme

2017

Day 4 GLMs - Poisson models for count data

The lecture introduced the basic concept of generalised linear models (GLM): we identify a link function that gives a good scale for fitting linear models (the *linear predictor*) and evaluate the fit of that model on the original data using an appropriate statistical distribution (the *error structure*).

In practice, this is very similar to using a linear model -- the key thing to be aware of is when we're using the scale of the linear predictor and when we're using the scale of the original data. The practical will give a walk through with one dataset and then provide examples to try.

Species richness on the Galapagos Islands

This dataset is on the number of plant species found on the Galapagos Islands (M. P. Johnson and P. H. Raven (1973) `Species number and endemism: The Galapagos Archipelago revisited' Science, 179, 893-895). It records the total number and number of endemic species along with information on the size and maximum elevation of the island and position in the archipelago. For more information, see the R package *'faraway'*.

We're going to use it for a very simple GLM analysis -- is there a relationship between the area of the island and the number of plant species found there?

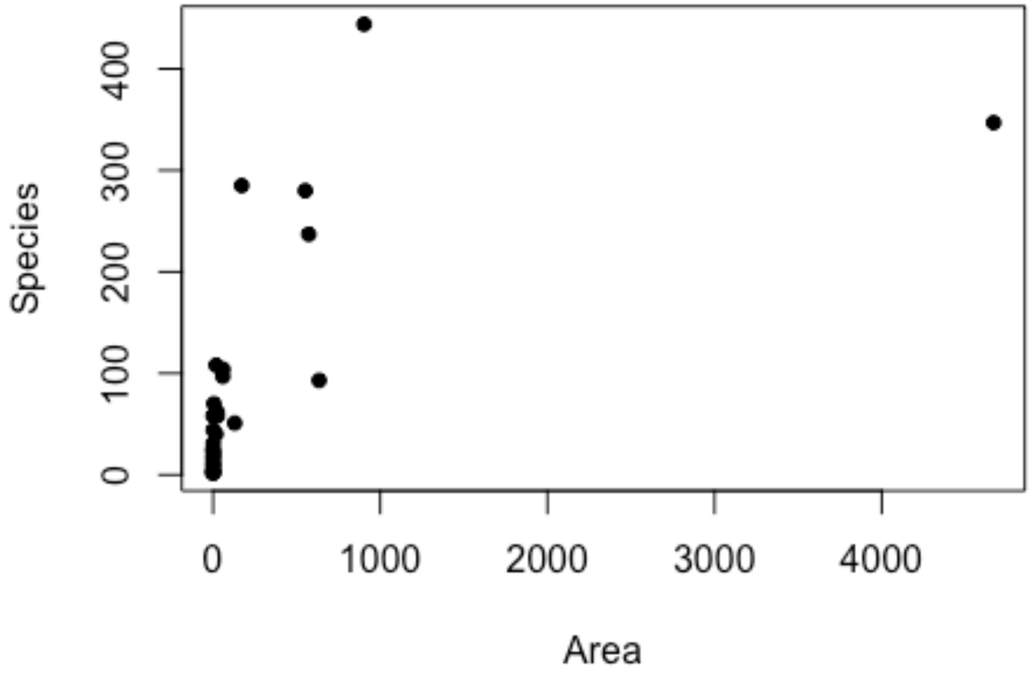
Before we begin, we clear our workspace. Never forget!

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

```
gala <- read.table("gala.txt", header=T)
str(gala)
```

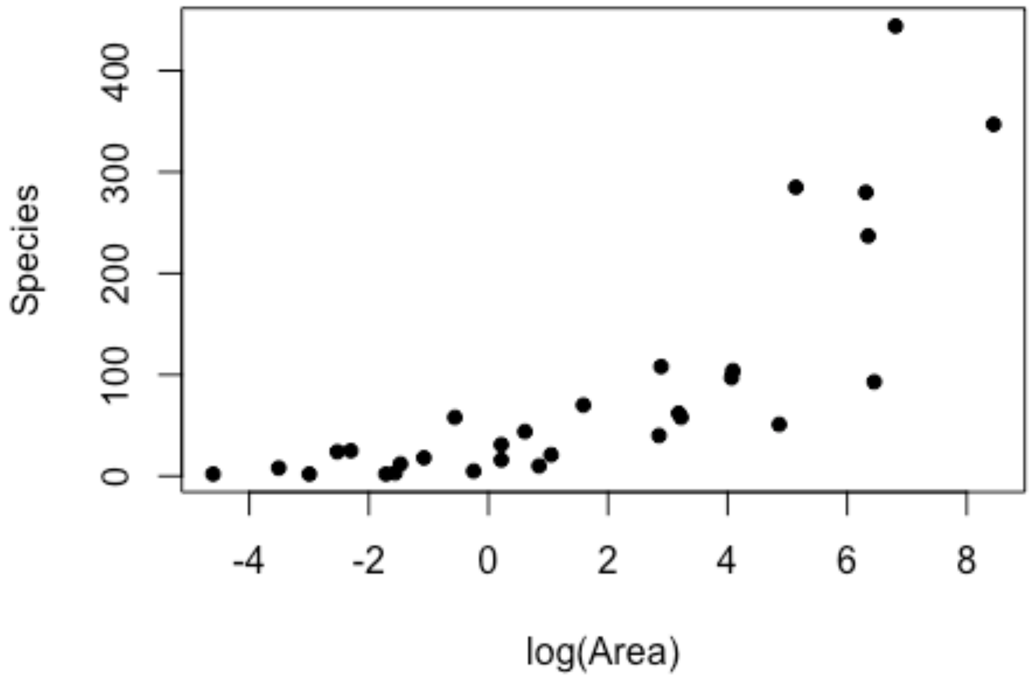
```
## 'data.frame':    30 obs. of  7 variables:
##  $ Species  : int  58 31 3 25 2 18 24 10 8 2 ...
##  $ Endemics : int  23 21 3 9 1 11 0 7 4 2 ...
##  $ Area     : num  25.09 1.24 0.21 0.1 0.05 ...
##  $ Elevation: int  346 109 114 46 77 119 93 168 71 112 ...
##  $ Nearest  : num  0.6 0.6 2.8 1.9 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 ...
```

```
plot(Species ~ Area , data=gala, pch=19, cex=0.8)
```



It seems to make sense to log-transform the area variable here. Just to be clear: this is a transformation that we make because the data is easier to handle and easier to visualize, - this has nothing to do with the poisson model. Logging the response variable gives us:

```
plot(Species ~ log(Area) , data=gala, pch=19, cex=0.8)
```



We can see from the plot that there appears to be a very strong relationship. The problem is that the data is count data: there is increasing variance and the data is bounded below at zero.

In order to fit a GLM, the only thing we need to change is to use the **glm()** function instead of the **lm()** function and specify the error distribution using the **family** option. This option also sets the *link function* to be used. For Poission data, the log link is the default, so we can just say **poisson** but to be clear we can say *'family=poisson(link=log)'*.

```
gala$lgArea <- log(gala$Area)
galaMod <- glm(Species ~ lgArea, data=gala, family=poisson(link=log))
```

That's it! That wasn't too hard, was it?

The summary of the coefficients from the model is very similar to the linear model output

but doesn't include r^2. This can't be defined for a GLM, since the residual sums of squares don't make sense as a measure of model fit, but we can calculate the proportion of the null deviance explained, which does a similar job.

```
summary(galaMod)

##
## Call:
```

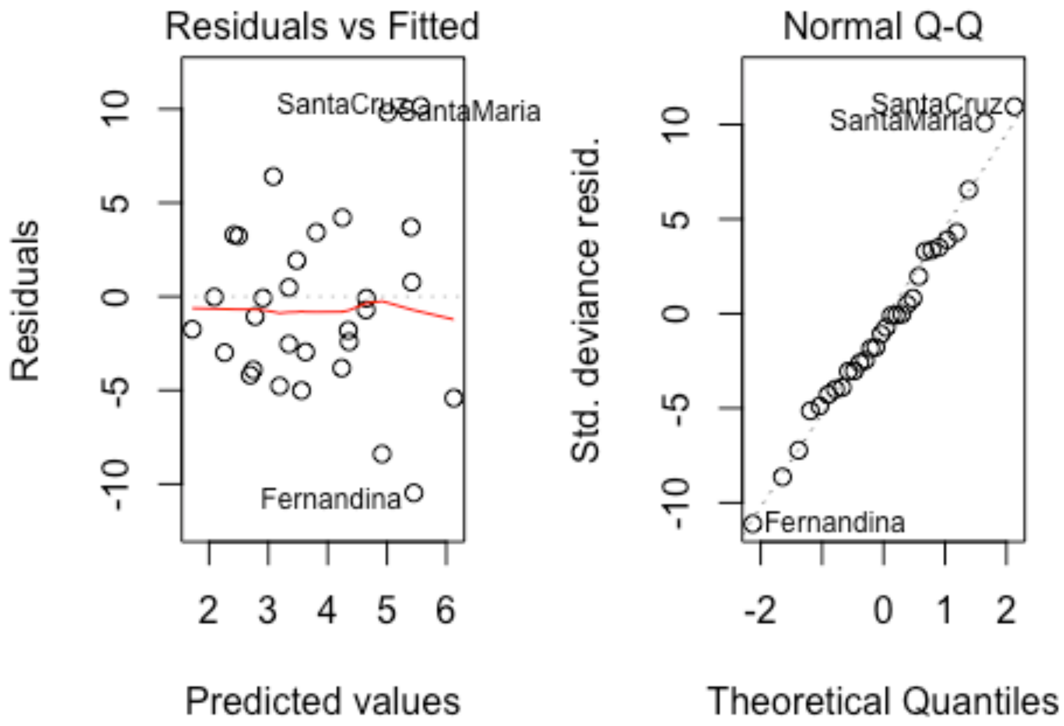
```
## glm(formula = Species ~ lgArea, family = poisson(link = log),
##     data = gala)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -10.4688   -3.6073   -0.8874    2.9028   10.1517
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 3.273200   0.041663   78.56   <2e-16 ***
## lgArea      0.337737   0.007154   47.21   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 3510.73  on 29  degrees of freedom
## Residual deviance:  651.67  on 28  degrees of freedom
## AIC: 816.5
##
## Number of Fisher Scoring iterations: 5
```

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

```
## [1] 0.8143775
```

We still should examine the diagnostic plots -- these plots now use the deviance residuals, and should be still be normally distributed with constant variance.

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



The bit that can be confusing is in plotting the model. The coefficients are on the scale of the linear predictor, so plotting them over the data, which we never actually transform and which might have zeros, needs some attention. The easy approach for the actual fitted means in the model is to get 'predict()' to give us the predictions on the scale of the response. Use ?predict to find out what it does!

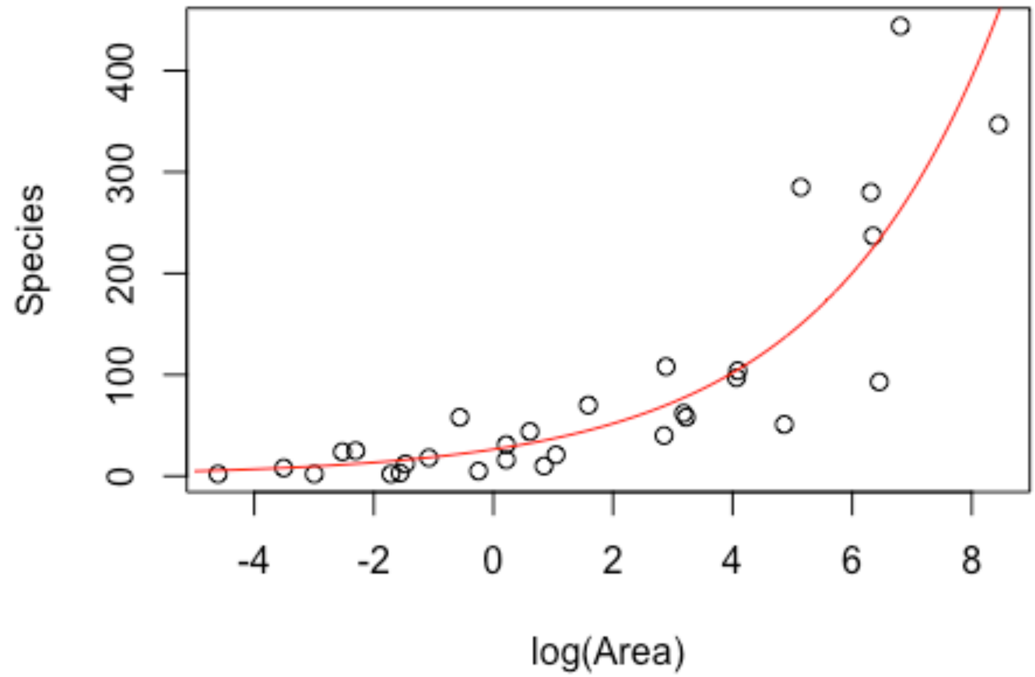
Now we can interpret the coefficient. The Intercept is 3.27, and it indicates the number of species in an area of 0. We will not try to interpret that one! The slope for the only covariate is 0.337. In a normal linear model we would think that means with an increase of 1 in area, we get 0.34 species more. But since this is a poisson model, we can't use this simple interpretation. We first have to back-transform the estimate:

```
exp(0.3377)

## [1] 1.40172
```

That means with each increase of area by one unit, we get 1.40% more species. It is relevant to remember that the slope is not constant in this model!

```
## null device
##           1
```

```r
# predict for a neat sequence of log area values
pred <- expand.grid(lgArea = seq(-5, 9, by=0.1))
head(pred)

##   lgArea
## 1   -5.0
## 2   -4.9
## 3   -4.8
## 4   -4.7
## 5   -4.6
## 6   -4.5

tail(pred)

##     lgArea
## 136    8.5
## 137    8.6
## 138    8.7
## 139    8.8
## 140    8.9
## 141    9.0

pred$fit <- predict(galaMod, newdata=pred, type='response')
head(pred)

##   lgArea      fit
## 1   -5.0 4.876917
## 2   -4.9 5.044442
## 3   -4.8 5.217722
## 4   -4.7 5.396953
## 5   -4.6 5.582341
## 6   -4.5 5.774098

# plot the logged data and the model lines

plot(Species ~ log(Area), data=gala)
lines(fit ~ lgArea, data=pred, col='red')
```



Amphibian roadkills in Portugal

The next dataset for you to try shows counts of the number of dead amphibians in 500 metre sections of a road in Portugal. There are a huge number of variables in the data frame measuring the local habitat characteristics. We won't use them all in the example.

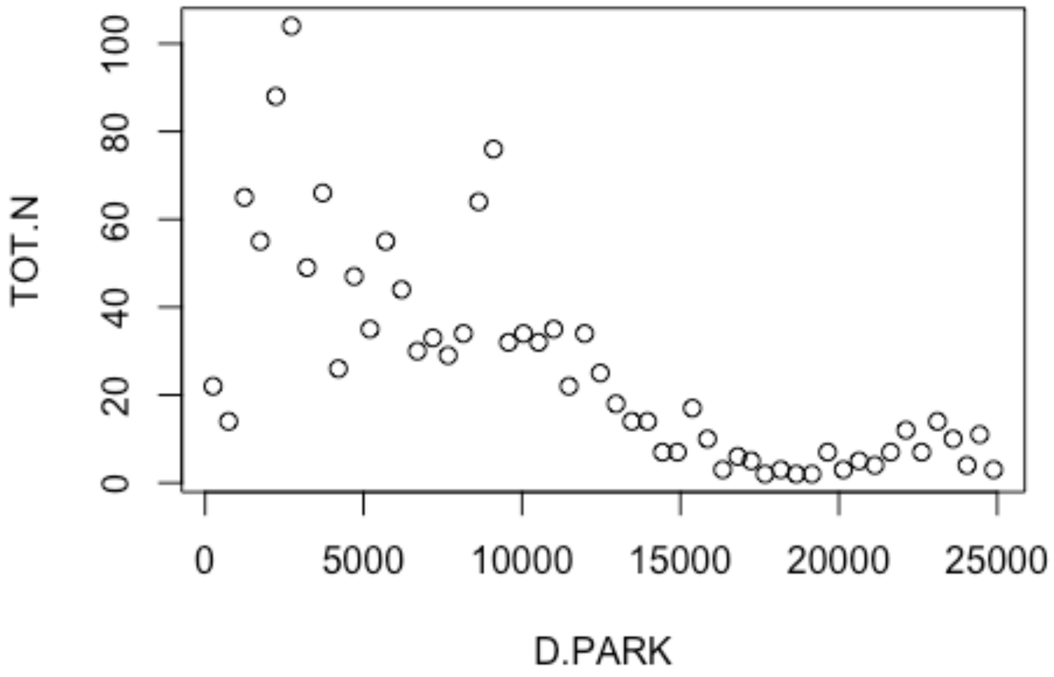
```
roadkill <- read.table("RoadKills.txt", header=T)
head(roadkill)

##   Sector      X      Y BufoCalamita TOT.N S.RICH OPEN.L  OLIVE MONT.S
## 1      1 260181 256546            5    22      3 22.684 60.333  0.000
## 2      2 259914 256124            1    14      4 24.657 40.832  0.000
## 3      3 259672 255688           40    65      6 30.121 23.710  0.258
## 4      4 259454 255238           27    55      5 50.277 14.940  1.783
## 5      5 259307 254763           67    88      4 43.609 35.353  2.431
## 6      6 259189 254277           56   104      7 31.385 17.666  0.000
##     MONT POLIC SHRUB  URBAN WAT.RES L.WAT.C L.D.ROAD L.P.ROAD D.WAT.RES
## 1  0.653 4.811 0.406  7.787   0.043   0.583 3330.189    1.975   252.113
## 2  0.161 2.224 0.735 27.150   0.182   1.419 2587.498    1.761   139.573
## 3 10.918 1.946 0.474 28.086   0.453   2.005 2149.651    1.250    59.168
## 4 26.454 0.625 0.607  0.831   0.026   1.924 4222.983    0.666   277.842
## 5 11.330 0.791 0.173  2.452   0.000   2.167 2219.302    0.653   967.808
## 6 43.678 0.054 0.325  2.730   0.039   2.391 1005.629    1.309   560.000
##   D.WAT.COUR   D.PARK N.PATCH  P.EDGE L.SDI
## 1    735.000  250.214     122 553.936 1.801
## 2    134.052  741.179      96 457.142 1.886
## 3    269.029 1240.080      67 432.360 1.930
## 4     48.751 1739.885      63 421.292 1.865
## 5    126.102 2232.130      59 407.573 1.818
## 6    344.444 2724.089      49 420.289 1.799
```

```
str(roadkill)
```

```
## 'data.frame':    52 obs. of  23 variables:
##  $ Sector      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ X           : int  260181 259914 259672 259454 259307 259189 259092
258993 258880 258767 ...
##  $ Y           : int  256546 256124 255688 255238 254763 254277 253786
253296 252809 252322 ...
##  $ BufoCalamita: int  5 1 40 27 67 56 27 37 8 16 ...
##  $ TOT.N       : int  22 14 65 55 88 104 49 66 26 47 ...
##  $ S.RICH      : int  3 4 6 5 4 7 7 7 7 6 ...
##  $ OPEN.L      : num  22.7 24.7 30.1 50.3 43.6 ...
##  $ OLIVE       : num  60.3 40.8 23.7 14.9 35.4 ...
##  $ MONT.S      : num  0 0 0.258 1.783 2.431 ...
##  $ MONT        : num  0.653 0.161 10.918 26.454 11.33 ...
##  $ POLIC       : num  4.811 2.224 1.946 0.625 0.791 ...
##  $ SHRUB       : num  0.406 0.735 0.474 0.607 0.173 ...
##  $ URBAN       : num  7.787 27.15 28.086 0.831 2.452 ...
##  $ WAT.RES     : num  0.043 0.182 0.453 0.026 0 0.039 0.114 0.224 0.177 0
...
##  $ L.WAT.C     : num  0.583 1.419 2.005 1.924 2.167 ...
##  $ L.D.ROAD    : num  3330 2587 2150 4223 2219 ...
##  $ L.P.ROAD    : num  1.975 1.761 1.25 0.666 0.653 ...
##  $ D.WAT.RES   : num  252.1 139.6 59.2 277.8 967.8 ...
##  $ D.WAT.COUR  : num  735 134.1 269 48.8 126.1 ...
##  $ D.PARK      : num  250 741 1240 1740 2232 ...
##  $ N.PATCH     : num  122 96 67 63 59 49 35 55 52 26 ...
##  $ P.EDGE      : num  554 457 432 421 408 ...
##  $ L.SDI       : num  1.8 1.89 1.93 1.86 1.82 ...
```

Now it's your turn to fit a Poisson GLM that predicts the number of road kills (*TOT.N*) as a function of distance to a nearby natural park (*D.PARK*). Go all the way and interpret the parameter estimates!

```
plot(TOT.N ~ D.PARK, data = roadkill)
```



Species richness in grassland plot

A third dataset includes records of plant species richness from 90 agricultural plots with differing soil pH (a three-level factor) and biomass (a continuous variable). Use this dataset to model whether species richness is predicted by soil pH and biomass and their interaction. Test whether we need the interaction with either AIC or logliklihood test.

```
species <- read.table("species.txt", header=T)
head(species)

##     pH   Biomass Species
## 1 high 0.4692972      30
## 2 high 1.7308704      39
## 3 high 2.0897785      44
## 4 high 3.9257871      35
## 5 high 4.3667927      25
## 6 high 5.4819747      29

str(species)

## 'data.frame':    90 obs. of  3 variables:
##  $ pH     : Factor w/ 3 levels "high","low","mid": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Biomass: num  0.469 1.731 2.09 3.926 4.367 ...
##  $ Species: int  30 39 44 35 25 29 23 18 19 12 ...
```

```
mfull<-glm(Species~pH*Biomass, data=species, family=poisson)
m2<-glm(Species~pH+Biomass, data=species, family=poisson)
require(lmtest)
```

```
lrtest(mfull,m2)
```

```
## Likelihood ratio test
## 
## Model 1: Species ~ pH * Biomass
## Model 2: Species ~ pH + Biomass
##   #Df  LogLik Df Chisq Pr(>Chisq)    
## 1   6 -251.20                        
## 2   4 -259.22 -2 16.04  0.0003288 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So the second model has a lower Loglikelihood than the full model (-259 vs -251 for the full model). The difference is statistically significant, which means the full model explains the data better.

We could also use the AIC:

```
AIC(mfull,m2)
```

```
##       df      AIC
## mfull  6 514.3913
## m2     4 526.4317
```

Which tells us the same - the difference between AIC is larger than 5, which is considered to be significantly different. The full model has the lower AIC, thus is the better model.

Try to interpret the model coefficients and the interaction!