Likelihood and AIC in linear models

Practical 2

David Orme

This practical handout is quite a bit shorter than the previous one[1]. It will introduce some simple tools for comparing the likelihood of models and then you can use those to explore using AIC and likelihood to compare models for the datasets from the last session

- How to get the log likelihood of a model
- Comparing models using AIC
- Automatic stepwise model selection using AIC

Again, create a new folder for this practical and write scripts for your code. Again, there is a short section at the end describing some of the analyses. We'll use the genome size in dragonflies data as worked example again.

Genome size in Odonata

The model from the last session gave us this analysis of covariance:

```
odonata <- read.csv('GenomeSize.csv')
```

```
str(odonata)
```

```
## 'data.frame': 100 obs. of  16 variables:
##  $ Suborder     : Factor w/ 2 levels "Anisoptera","Zygoptera": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Family      : Factor w/ 9 levels "Aeshnidae","Calopterygidae",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Species      : Factor w/ 100 levels "Aeshna canadensis",..: 1 2 3 4 5 6 8 17 46 53 ...
##  $ GenomeSize    : num  2.2 1.76 1.85 1.78 2 1.59 1.44 1.16 1.44 1.2 ...
##  $ GenomeSE      : num  NA 0.06 NA 0.1 NA NA NA NA NA NA ...
##  $ GenomeN       : int  1 4 1 2 1 1 1 1 1 1 ...
##  $ BodyWeight    : num  0.159 0.228 0.312 0.218 0.207 0.22 0.344 0.128 0.392 0.029 ...
##  $ TotalLength   : num  67.6 72 78.8 72.4 73 ...
##  $ HeadLength    : num  6.83 6.84 6.27 6.62 4.92 6.48 7.53 5.74 8.05 5.28 ...
##  $ ThoraxLength  : num  11.8 10.7 16.2 12.5 11.1 ...
##  $ AdbdomenLength: num  48.9 54.4 56.3 53.3 57 ...
##  $ ForewingLength: num  45.5 46 51.2 49.8 46.5 ...
##  $ HindwingLength: num  45.4 45.5 49.5 48.8 46 ...
##  $ ForewingArea  : num  370 411 461 469 382 ...
##  $ HindwingArea  : num  484 517 574 591 481 ...
##  $ MorphologyN   : int  2 3 1 2 1 1 4 1 1 1 ...
```

[1] Phew!

```r
odonataFull <- lm(log(BodyWeight) ~ log(GenomeSize) * Suborder, data=odonata)
summary(odonataFull)

##
## Call:
## lm(formula = log(BodyWeight) ~ log(GenomeSize) * Suborder, data = odonata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3243 -0.3225  0.0073  0.3962  1.4976
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      -2.3995     0.0848  -28.31  < 2e-16 ***
## log(GenomeSize)                   1.0052     0.2237    4.49  2.0e-05 ***
## SuborderZygoptera                -2.2489     0.1354  -16.61  < 2e-16 ***
## log(GenomeSize):SuborderZygoptera -2.1492     0.4619   -4.65  1.1e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.65 on 94 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.755,Adjusted R-squared:  0.747
## F-statistic: 96.5 on 3 and 94 DF,  p-value: <2e-16

anova(odonataFull)

## Analysis of Variance Table
##
## Response: log(BodyWeight)
##                           Df Sum Sq Mean Sq F value  Pr(>F)
## log(GenomeSize)            1    1.1     1.1    2.71     0.1
## Suborder                   1  112.0   112.0  265.13 < 2e-16 ***
## log(GenomeSize):Suborder   1    9.1     9.1   21.65 1.1e-05 ***
## Residuals                 94   39.7     0.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Comparing models with ANOVA

We know that this is the minimum adequate model but the possible models that we could have are:

```r
odonataFull  <- lm(log(BodyWeight) ~ log(GenomeSize) * Suborder, data=odonata)
odonataNoInt <- lm(log(BodyWeight) ~ log(GenomeSize) + Suborder, data=odonata)
odonataBW    <- lm(log(BodyWeight) ~ log(GenomeSize), data=odonata)
odonataSub   <- lm(log(BodyWeight) ~ Suborder, data=odonata)
odonataNull <- lm(log(BodyWeight) ~ 1, data=odonata)
```

In the past, we've used an analysis of variance table to compare the fit of models:

```r
anova(odonataFull, odonataNoInt, odonataBW, odonataSub, odonataNull)

## Analysis of Variance Table
```

```
##
## Model 1: log(BodyWeight) ~ log(GenomeSize) * Suborder
## Model 2: log(BodyWeight) ~ log(GenomeSize) + Suborder
## Model 3: log(BodyWeight) ~ log(GenomeSize)
## Model 4: log(BodyWeight) ~ Suborder
## Model 5: log(BodyWeight) ~ 1
##   Res.Df   RSS Df Sum of Sq     F  Pr(>F)
## 1     94  39.7
## 2     95  48.8 -1      -9.1  21.6 1.1e-05 ***
## 3     96 160.8 -1    -112.0 265.1 < 2e-16 ***
## 4     96  51.6  0     109.2
## 5     97 162.0 -1    -110.3 261.3 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

That works (and tells us not to drop anything) but there is the issue of comparing the two models with only body mass (odonoataBW) and only suborder (odonoataSub). Analysis of variance can't test whether the change in residual sums of squares is unusual given the change in degrees of freedom, because there isn't any change in degress of freedom.

Getting likelihoods

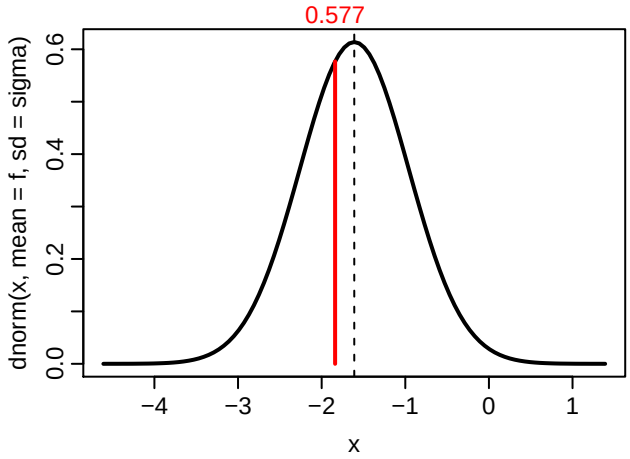
This isn't difficult – there is a simple function to get the log likelihood from a linear model:

```
logLik(odonataFull)

## 'log Lik.' -94.8 (df=5)
```

Not only does that give us the log likelihood, it also reports the degrees of freedom needed to calculate the log likelihood. This is *one more than the number of coefficients*: in order to calculate the likelhood, we also have to estimate the variance in the residuals.

To link back to the lecture, that number is the sum of the logged probabilities of each point under the model. To take an example point from the model: log genome size (x=0.7885), log body mass (y=−1.8390) with a predicted value from the model of f=−1.6100. Given the standard deviation of the residuals in the model ($sigma$=0.65), the distribution of residuals for the point and the density of the observed points are:

```
x <- 0.7885
sigma <- 0.65
y <- -1.839
f <- -1.61
dy <- dnorm(y, mean=f, sd=sigma)
print(dy)

## [1] 0.577
```



We can therefore calculate the total log likelihood like this:

```
library(MASS)
# get the predictions
f <- fitted(odonataFull)
# get the observations
r <- resid(odonataFull)
y <- f + r
# get the standard deviation of the residuals
residDist <- fitdistr(r, 'normal')
sigma <- residDist$estimate[2]

# putting that all together
ll <- sum(dnorm(y, mean=f, sd=sigma, log=TRUE))
print(ll)

## [1] -94.8
```

Getting and comparing AIC

It is similarly easy to get the AIC for a model, which is -2 times the log likelihood plus 2 times the number of parameters (including the estimated standard error).

```
AIC(odonataFull)

## [1] 200

-2 * ll + 2 * 5

## [1] 200
```

But **AIC**() has more to it than just calculating the AIC for one model. You can also use it to compare models:

```
AIC(odonataFull, odonataNoInt, odonataBW, odonataSub, odonataNull)

##              df AIC
## odonataFull   5 200
## odonataNoInt  4 218
## odonataBW     3 333
```

```
## odonataSub    3 221
## odonataNull   2 331
```

We're looking for the lowest AIC, so this shows us that the full model is the best one. No surprises there, but AIC allows us to compare models with the same degrees of freedom: we can see that the model with only body mass is much worse than the model with only suborder. In fact, it is worth than the null model!

Simplifying models using AIC

The function **step**() works like a bit like **drop1**(): it looks at all possible single terms to drop from a model and tests how those affect the model. However, it goes a bit further – if removing any terms lowers AIC and therefore making it better, it finds which of those terms gives the best improvement and then does it all over again. It keeps doing this until it gets to point where removing any term makes the AIC bigger[2] and then gives back the final model.

At each stage, just like **drop1**(), you get a quick summary of what options were considered. We'll test it on the dragonfly model first, to show the output.

```
stepModel <- step(odonataFull)

## Start:  AIC=-80.6
## log(BodyWeight) ~ log(GenomeSize) * Suborder
##
##                             Df Sum of Sq  RSS   AIC
## <none>                                   39.7 -80.6
## - log(GenomeSize):Suborder  1      9.14 48.8 -62.2
```

It can only remove one thing (the interaction) and that makes the AIC smaller.[3]

That isn't a particularly interesting example, so let's look at how it works for the mammal genome size model. We first load and create the model

```
mammals <- read.csv('MammalData.csv')
```

```
model <- lm(log(meanCvalue) ~  (log(LitterSize) + log(AdultBodyMass_g) +
           TrophicLevel + GroundDwelling)^2, data = mammals)
```

Now let's look at the first iteration only, to see some more complex output:

```
stepMammal <- step(model, steps = 1)

## Start:  AIC=-767
## log(meanCvalue) ~ (log(LitterSize) + log(AdultBodyMass_g) + TrophicLevel +
##     GroundDwelling)^2
##
##                                       Df Sum of Sq  RSS  AIC
## - TrophicLevel:GroundDwelling          2     0.045 8.70 -770
## - log(AdultBodyMass_g):TrophicLevel    2     0.056 8.71 -770
## - log(LitterSize):log(AdultBodyMass_g) 1     0.016 8.67 -769
```

[2] Which is bad

[3] You may have noticed that the starting AIC is different to the one we calculated above. There is more than one way of calculating AIC for linear models – the key thing is that although the absolute values have changed the differences between models hasn't.

```
## - log(LitterSize):TrophicLevel          2     0.099 8.75 -769
## <none>                                         8.65 -767
## - log(LitterSize):GroundDwelling        1     0.662 9.32 -752
## - log(AdultBodyMass_g):GroundDwelling   1     0.867 9.52 -747
##
## Step:  AIC=-770
## log(meanCvalue) ~ log(LitterSize) + log(AdultBodyMass_g) + TrophicLevel +
##    GroundDwelling + log(LitterSize):log(AdultBodyMass_g) + log(LitterSize):TrophicLevel +
##      log(LitterSize):GroundDwelling + log(AdultBodyMass_g):TrophicLevel +
##      log(AdultBodyMass_g):GroundDwelling
```

There are six possible model simplifications: the six two-way interactions (a:b, a:c, a:d, b:c, b:d and c:d). Models dropping each on of those in turn are all compared together to the original model. Although the top four models all have very similar AIC, the top one is the one that gets dropped.

Now let's run it the whole way through:

```
stepMammal <- step(model)

## Start:  AIC=-767
## log(meanCvalue) ~ (log(LitterSize) + log(AdultBodyMass_g) + TrophicLevel +
##      GroundDwelling)^2
##
##                                      Df Sum of Sq  RSS  AIC
## - TrophicLevel:GroundDwelling         2     0.045 8.70 -770
## - log(AdultBodyMass_g):TrophicLevel   2     0.056 8.71 -770
## - log(LitterSize):log(AdultBodyMass_g)  1     0.016 8.67 -769
## - log(LitterSize):TrophicLevel          2     0.099 8.75 -769
## <none>                                         8.65 -767
## - log(LitterSize):GroundDwelling        1     0.662 9.32 -752
## - log(AdultBodyMass_g):GroundDwelling   1     0.867 9.52 -747
##
## Step:  AIC=-770
## log(meanCvalue) ~ log(LitterSize) + log(AdultBodyMass_g) + TrophicLevel +
##    GroundDwelling + log(LitterSize):log(AdultBodyMass_g) + log(LitterSize):TrophicLevel +
##      log(LitterSize):GroundDwelling + log(AdultBodyMass_g):TrophicLevel +
##      log(AdultBodyMass_g):GroundDwelling
##
##                                      Df Sum of Sq  RSS  AIC
## - log(AdultBodyMass_g):TrophicLevel   2     0.017 8.71 -774
## - log(LitterSize):TrophicLevel          2     0.079 8.78 -772
## - log(LitterSize):log(AdultBodyMass_g)  1     0.009 8.71 -772
## <none>                                         8.70 -770
## - log(LitterSize):GroundDwelling        1     0.723 9.42 -753
## - log(AdultBodyMass_g):GroundDwelling   1     0.883 9.58 -749
##
## Step:  AIC=-774
## log(meanCvalue) ~ log(LitterSize) + log(AdultBodyMass_g) + TrophicLevel +
##    GroundDwelling + log(LitterSize):log(AdultBodyMass_g) + log(LitterSize):TrophicLevel +
##      log(LitterSize):GroundDwelling + log(AdultBodyMass_g):GroundDwelling
##
##                                      Df Sum of Sq  RSS  AIC
## - log(LitterSize):log(AdultBodyMass_g)  1     0.005 8.72 -776
```

```
## - log(LitterSize):TrophicLevel          2     0.093 8.81 -775
## <none>                                         8.71 -774
## - log(LitterSize):GroundDwelling        1     0.710 9.42 -757
## - log(AdultBodyMass_g):GroundDwelling   1     0.954 9.67 -751
##
## Step:  AIC=-776
## log(meanCvalue) ~ log(LitterSize) + log(AdultBodyMass_g) + TrophicLevel +
##     GroundDwelling + log(LitterSize):TrophicLevel + log(LitterSize):GroundDwelling +
##     log(AdultBodyMass_g):GroundDwelling
##
##                                      Df Sum of Sq  RSS  AIC
## - log(LitterSize):TrophicLevel        2     0.093 8.81 -777
## <none>                                            8.72 -776
## - log(LitterSize):GroundDwelling      1     0.881 9.60 -755
## - log(AdultBodyMass_g):GroundDwelling 1     1.053 9.77 -750
##
## Step:  AIC=-777
## log(meanCvalue) ~ log(LitterSize) + log(AdultBodyMass_g) + TrophicLevel +
##    GroundDwelling + log(LitterSize):GroundDwelling + log(AdultBodyMass_g):GroundDwelling
##
##                                      Df Sum of Sq   RSS  AIC
## <none>                                            8.81 -777
## - TrophicLevel                        2     0.196  9.01 -776
## - log(LitterSize):GroundDwelling      1     0.957  9.77 -754
## - log(AdultBodyMass_g):GroundDwelling 1     1.207 10.02 -748
```

This takes a while to look through but the last section shows that all three possible simplifiations (dropping trophic level or either of the arboreality interactions) make the AIC larger than the current best model. So we stop there and the function returns the final model:

```
anova(stepMammal)

## Analysis of Variance Table
##
## Response: log(meanCvalue)
##                                     Df Sum Sq Mean Sq F value  Pr(>F)
## log(LitterSize)                      1   0.99   0.989   26.04 6.9e-07 ***
## log(AdultBodyMass_g)                 1   3.03   3.032   79.81 < 2e-16 ***
## TrophicLevel                         2   0.48   0.239    6.29  0.0022 **
## GroundDwelling                       1   0.11   0.110    2.91  0.0895 .
## log(LitterSize):GroundDwelling       1   0.25   0.251    6.60  0.0108 *
## log(AdultBodyMass_g):GroundDwelling  1   1.21   1.207   31.77 5.0e-08 ***
## Residuals                          232   8.81   0.038
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

It isn't the same as the one I came up with using **drop1**(): there are a lot of possible models and it isn't always clear which is best. Also step using AIC tends to be more permissive than anova – it often includes rather marginal terms.

Another example dataset

The next dataset looks at the life history of the Galliformes: the game birds such as pheasant, quail and partridge. It contains the body mass, range size, elevational range and clutch size of these bird species. It also contains information on their IUCN threat status and we'll come to look at that variable later.

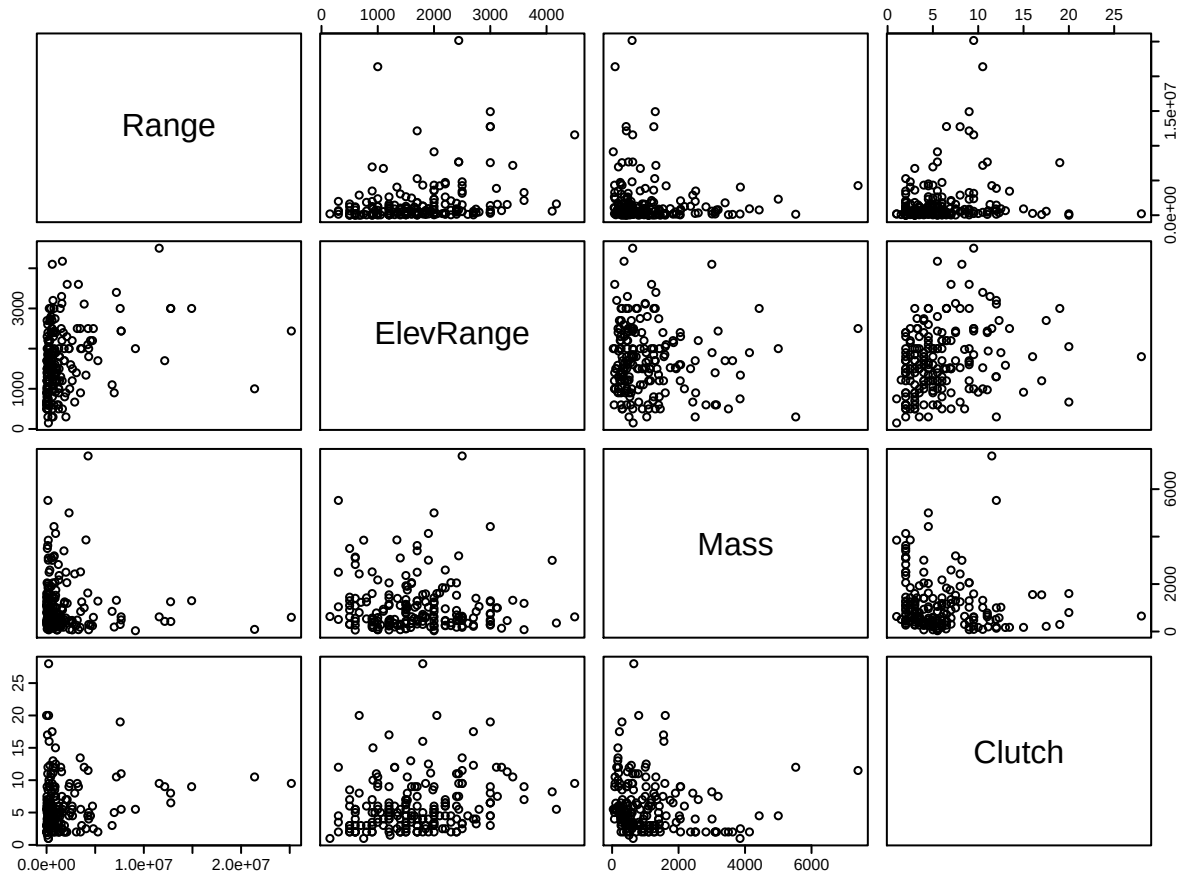
For now, we'll look to see if the range size of the species is predicted by their body mass, clutch size and elevational range. We will first load the data and fit the most complex model.

```
galliformes <- read.csv('galliformesData.csv')
```

```
# drop rows of data that contain missing values
galliformes <- na.omit(galliformes)
str(galliformes)

## 'data.frame': 194 obs. of  8 variables:
##  $ Family   : Factor w/ 6 levels "Cracidae","Megapodiidae",..: 1 2 5 5 5 5 5 5 5 5 ...
## $ CName    : Factor w/ 268 levels "Aceh Pheasant",..: 251 249 71 10 66 193 203 5 173 186 ...
##  $ SName    : Factor w/ 268 levels "Aburria aburri",..: 1 2 4 7 8 9 10 11 12 13 ...
##  $ Status04 : Factor w/ 5 levels "1 (LC)","2 (NT)",..: 2 1 3 1 1 1 1 1 1 1 ...
## $ Range    : int  778240 205731 202631 1542970 11577357 446540 345484 437252 163851 901099 ...
##  $ Mass     : num  1423 1600 1418 461 619 ...
##  $ Clutch   : num  3.25 20 2.5 11.3 9.5 ...
##  $ ElevRange: int  2000 2050 1200 3300 4500 2400 2700 3000 2200 1300 ...
##  - attr(*, "na.action")=Class 'omit'  Named int [1:74] 3 5 6 14 17 20 21 24 26 28 ...
##   .. ..- attr(*, "names")= chr [1:74] "3" "5" "6" "14" ...
```

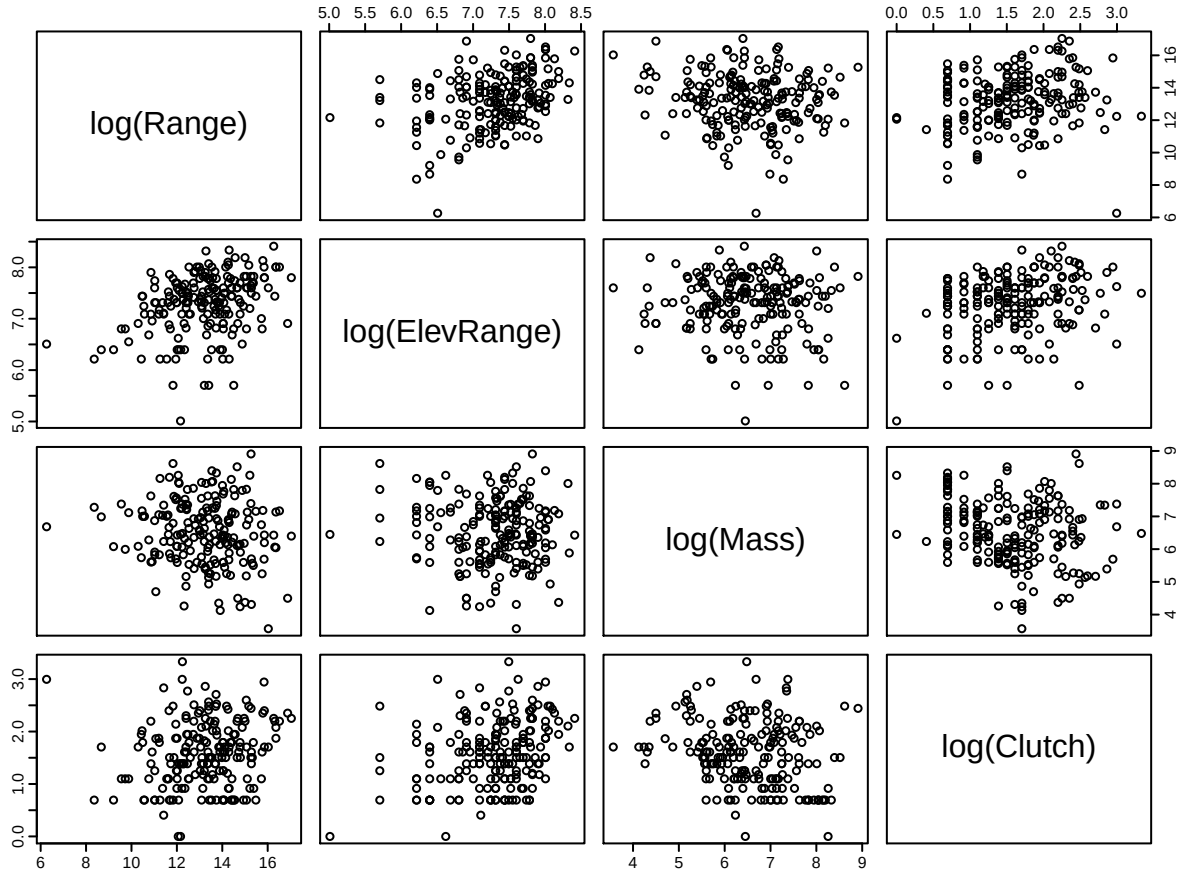
Before we go any further, we'll use the pairs function to check the data. There is a neat trick that allows us to use the model formula to create exactly the plots we want to see for the model:

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



That doesn't look very good, so we'll again use log transformations to see if that helps.

```
pairs(log(Range) ~ log(ElevRange)*log(Mass)*log(Clutch), data=galliformes)
```



That's much better – so now we'll get a new starting model.

```
galliModel <- lm(log(Range) ~ log(ElevRange)*log(Mass)*log(Clutch), data=galliformes)
```

Now use **step**() and **drop1**() to simplify this model. You'll find that the AIC simplification leaves you with rather a complex model where the terms aren't very signficant, which is where you may want to use ANOVA to simplify further.

Galliform range size

Methods

I fitted a linear model predicting the log range size (km^2) of 194 galliform bird species as a function of their log body mass in grams, log clutch size and log elevational range. I started from a fully factorial model including the three explanatory variables. I used AIC step fitting to simplify the initial model and then used F tests to further simplify the model by testing for marginally significant terms. I used residual diagnostic plots to assess model suitability.

Results

The minimum adequate model shows that elevational range strongly predicts range size ($F_{1,190} = 27.7, p < 0.0001$) and that clutch size ($F_{1,190} = 1.1, p = 0.3$) interacts with elevational range ($F_{1,190} = 5.25, p = 0.023$). There is considerable variation around the model ($r^2 = 0.14$, Figure 1), the model coefficients of which are given in Table 1.

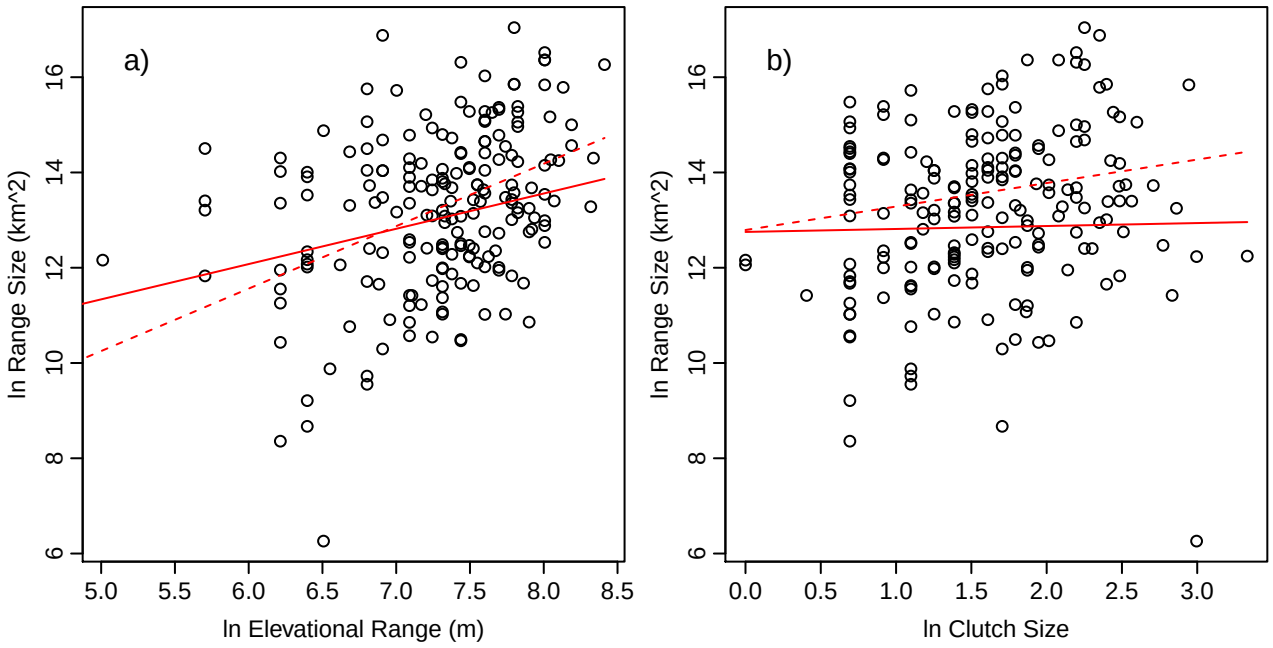


Figure 1: *Variation in galliform range size with a) species' elevational range and b) clutch size. Predictions from the linear model are shown in red for : a) the first (solid line) and third (dashed line) quartile values of log clutch size; b) the first (solid line) and third (dashed line) quartile values of log elevational range.*

Table 1: Model coefficients, standard errors and signficance tests for a linear model of galliform range size as a function of clutch size and elevational gradient.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	12.3479	3.2201	3.83	0.0002
log(ElevRange)	0.0576	0.4519	0.13	0.8986
log(Clutch)	-4.2935	1.9702	-2.18	0.0305
log(ElevRange):log(Clutch)	0.6219	0.2714	2.29	0.0230