Model Selection and Multimodel Inference

Scott creel

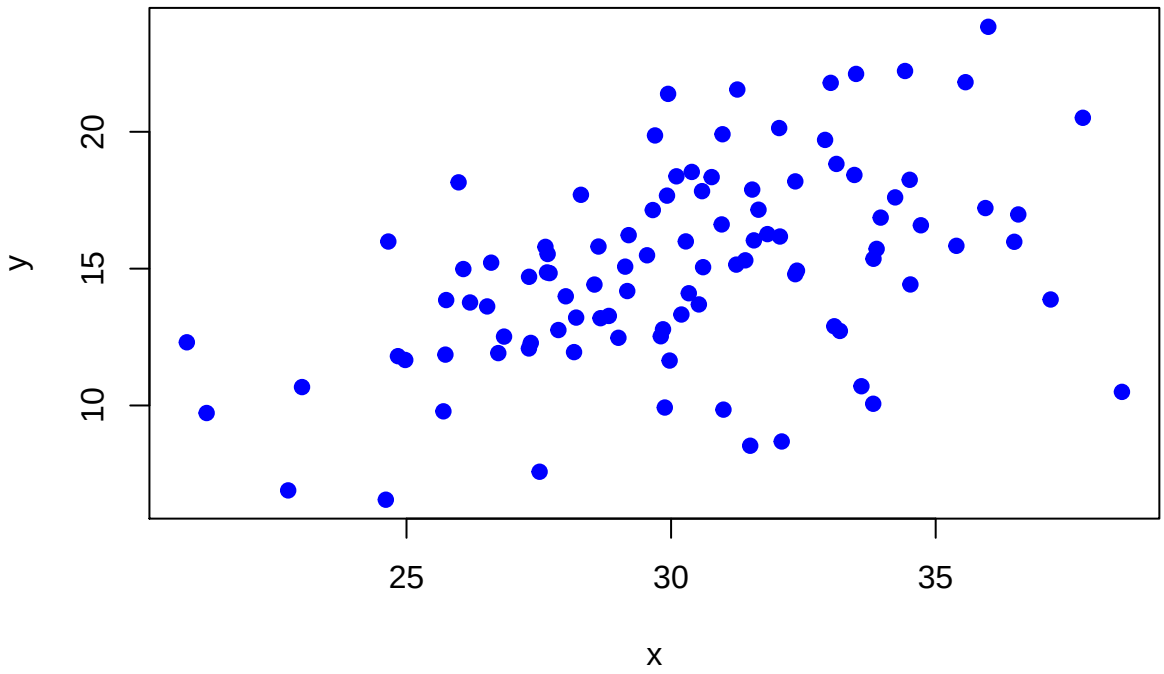
Thursday, September 11, 2014

The last R Exercise introduced **generalized linear models** and how to fit them in R using the **glm()** function. That exercise briefly noted that you can use glm() with family = gaussian (gaussian means 'normally distributed') to fit the same OLS regression model that lm() would fit.

An example to show the relationship of lm() to glm(family = gaussian)

First, create some simulated data with 100 observations of y and x, with an underlying relationship between the two variables that is affect by (normally-distributed) random variation.

```
x <- (rnorm(n = 100, mean = 30, sd = 4))
y <- rnorm(n= 100, mean = 0.5, sd = 0.1) * x
par(mfrow = c(1,1))
plot(x,y, pch = 19, col = 'blue')
```



Now, fit a linear regression model to the data with lm().

```
mod1 <- lm(y~x)
summary(mod1)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.356 -1.495  0.038  1.908  6.476
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.1473     2.6969    0.43     0.67
## x             0.4596     0.0886    5.19  1.1e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.18 on 98 degrees of freedom
## Multiple R-squared:  0.216,  Adjusted R-squared:  0.208
## F-statistic: 26.9 on 1 and 98 DF,  p-value: 1.14e-06
```

Using lm() fits the model by Ordinary Least Squares. As should be familiar from the earlier exercises, the summary() function provides:

- an estimate of the effect of x on y (a regression coefficient),
- the standard error of that regression coefficient
- a t-statistic testing whether the slope differs from zero – in other words, whether there is a detectable relationship between y and x, with a given level of certainty (as measured by the P-value) In this case, there is a high degree of certainty that y is related to x.
- You also see the coefficient of determination (R-squared adjusted for degrees of freedom), which tells you what percentage of the variation in y can be explained by knowing the value of x. In this case, R2-adjusted is about 25%. (Because I used the rnorm() function to generate a random sample of data, the exact number will change each time the code is run but the basic pattern will stay the same, with a slope about 0.5, which is the mean for the population from which the sample is drawn)

Now fit a linear regression with glm(family = gaussian).

```
mod2 <- glm(y~x, family = gaussian)
summary(mod2)
```

```
##
## Call:
## glm(formula = y ~ x, family = gaussian)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -8.356  -1.495   0.038   1.908   6.476
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.1473     2.6969    0.43     0.67
## x             0.4596     0.0886    5.19  1.1e-06 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 10.09)
##
##     Null deviance: 1260.60  on 99  degrees of freedom
## Residual deviance:  988.79  on 98  degrees of freedom
## AIC: 518.9
##
## Number of Fisher Scoring iterations: 2
```

This summary shows that we obtain **exactly** the same regression coefficient, standard error, and significance test for the effect of x on y. The difference is in the output at the bottom.

glm() fits a model by **maximum likelihood (ML)**, so the assessment of the model's performance is different. Using maximum likelihood, You do not get a coefficient of determination. Instead, you get a **residual deviance** value and an **AIC** value. What are these?

Fitting a model by ML, the likelihood that is maximized is the likelihood of the estimated parameters (β_0 and β_1), given the observed data:

$$\pounds(parameters|data) = \pounds(\beta_0, \beta_1|data)$$

In practice, maximum likelihood estimation is actually done by **minimizing the residual deviance**.

residual deviance $= -2 * ln(\pounds(parameters|data))$

Often the word residual will be omitted and it's just called **deviance**. The deviance measures the **unexplained variation**, like the residuals in OLS regression, so we want it to be as small as possible. The glm() function basically checks all possible regression lines (all combinations of β_0 and β_1), calculates a deviance each time, and picks the model with the lowest deviance.

- Small deviance = good model.
- Large deviance = bad model.

Don't make the following mistake!

The coefficient of determination is an absolute measure, so you can compare R2 values for models fit to different data sets. The deviance is a **relative** measure of how well a model fits a data set. This means that you can compare models by comparing deviance **only if they are fit to the same data**. (This makes sense when you look at the likelihood statement – it is the likelihood of the parameters **given the data**). In practice, if you have a data set with **missing observations for some of the variables but not others**, you **can not compare models** by comparing deviance (or AIC scores, as discussed just below).

Model selection using AIC scores

AIC stands for **Akaike's Information Criterion**. The AIC score for a model is equal to the residual deviance plus two times the number of parameters (K) in the model.

$$AIC = -2 * ln(\pounds(parameters|data)) + 2K$$

A model fits well if it has a small deviance, so when comparing models, the one with the **smaller AIC score is better supported by the data**. A model that produces a small deviance with few parameters

is more **parsimonious** than a model that yields the same deviance with more parameters. By adding 2K to the deviance, AIC score penalize models that require a lot of parameters to achieve a good fit to the data. In other words, AIC scores favor simpler models, relative to the conclusions that you'd draw by simply comaring the deviance for a pair of models. (All of that is also true of R2 adjusted for degrees of freedom. R2 is a measure of the variance in y explained by x. Adjusting for the degrees of freedom accomplishes the same basic result as penalizing an AIC score by 2K. However, model selection on the basis of AIC scores is currently more widely practiced than model selection on the basis of R2-adjusted.)

AIC scores are often corrected for small sample size (AIC_c). The equation for this is in CWP CH. 2.

An example of model selection using AIC scores in R

As emphasized in CWP CH.2, strong inferences can be drawn when you compare a set of competing hypotheses to see which has most support from the data. Each regression model that you fit represents a hypothesis about the factors that affect your dependent variable. By formulating a set of hypotheses and fitting a set regression models that corresponds to each hypothesis, you can use AIC scores to examine the support that each model received from the data.

Here is a worked example in R, using a data set on behavioral responses to predation risk:

You will have to set the working directory to the location where you save the data file kenyaherdsize3.

```
rm(list=ls(all=TRUE))
kenya.herdsize = read.table("kenyaherdsize3.txt",
          header = TRUE, sep = ",", fill= TRUE, dec = ".")
```

The input file kenyaherdsize3 differs from kenyaherdsize2 (which we used in R Exercise 3A) by having all observations with missing data for any of the variables deleted. It has 348 observations of 41 variables. kenyaherdsize2 would be more useful for some purposes, but would not allow for valid comparison of AIC scores for different models. The variables we'll use here are a subset of the ones you examined previously:

GroupSize - size of the ungulate herd

Species - species of the ungulate herd

DistPred - distance from the herd to the nearest predator in kilometers

BushWoodGrass - vegetation type as a categorical variable with three levels (B =bush ,W = woodland, G = grassland)

HabOpen.Close - vegetation type as a simpler categorical variable with only two levels (O = open, c = closed)

Model selection using AIC scores requires the following steps:

1. Develop a set of hypotheses about the factors that affect herd size.
2. Collect the data required.
3. Fit a model for each hypothesis in set.
4. Compare models using AIC scores
5. Use the information from step 4 to estimate regression coefficients by using 'model-averaging'.

Here is a simple set of four hypotheses to evaluate:

Model A: Wildebeest group size is affected by differences in vegetation structure in woodland, bushland and grassland Model B: Wildebeest group size is affected by vegetation structure in a simpler manner, depending only on whether the habitat is open or closed. Model C: Wildebeest group size is affected by the proximity of

predators Model D: Wildebeest group size is affected by proximity of predators and by the differences among woodland, bushland and grassland.

Fit the four models using glm()

```
wildebeest.only <-  subset(reduced.data, Species == 'Wildbst',
                 select = c(GroupSize, BushWoodGrass, HabOpen.Close, DistPred))

modA <- glm(formula = GroupSize ~ BushWoodGrass, data = wildebeest.only)

modB <- glm(formula = GroupSize ~ HabOpen.Close, data = wildebeest.only)

modC <- glm(formula = GroupSize ~ DistPred, data = wildebeest.only)

modD <- glm(formula = GroupSize ~ DistPred + BushWoodGrass, data = wildebeest.only)
```

Compare AIC scores for the 4 models using functions in the MuMIn (Multi-Model Inference) package:

```
library(MuMIn)
Cand.mods <- list(modA, modB, modC, modD)
aictab <- model.sel(Cand.mods)
aictab
```

```
## Model selection table
##   (Int)  BWG HbO.Cls DsP   df logLik AICc  delta weight
## 3  5.010             6.875 3  -213.9 434.3 0.00  0.602
## 2 10.110     +             3  -215.1 436.7 2.41  0.180
## 4  1.585 +           7.002 5  -213.2 437.6 3.36  0.112
## 1  8.100 +                 4  -214.4 437.7 3.46  0.107
```

The output shows a summary of the structure of each model, the number of parameters, AIC and ΔAIC scores, and AIC weights (w). The interpretation of these is covered in CWP CH. 2 pages 24-27.

The function below reformats and re-organizes the AIC table so that it is a little easier to read, with the models ordered from best-supported to worst-supported, as is usually done for AIC tables in publications.

```
print.data.frame(aictab,digits=2)
```

```
##   (Intercept) BushWoodGrass HabOpen.Close DistPred df logLik AICc delta
## 3         5.0          <NA>          <NA>      6.9  3   -214  434   0.0
## 2        10.1          <NA>             +       NA  3   -215  437   2.4
## 4         1.6             +          <NA>      7.0  5   -213  438   3.4
## 1         8.1             +          <NA>       NA  4   -214  438   3.5
##   weight
## 3   0.60
## 2   0.18
## 4   0.11
## 1   0.11
```

Obtain estimates of the regression coefficients that are averaged across models, with each value weighted by the AIC weight for the model that contained it:

```
x <-model.avg(Cand.mods, beta = TRUE, revised.var = TRUE)
summary(x, digits = 3)
```

```
##
## Call:
## model.avg.default(object = Cand.mods, beta = TRUE, revised.var = TRUE)
##
## Component models:
##    df logLik  AICc Delta Weight
## 2   3 -213.9 434.3  0.00   0.60
## 3   3 -215.1 436.7  2.41   0.18
## 12  5 -213.2 437.6  3.36   0.11
## 1   4 -214.4 437.7  3.46   0.11
##
## Term codes:
## BushWoodGrass      DistPred HabOpen.Close
##             1             2             3
##
## Model-averaged coefficients:
##                Estimate Std. Error Adjusted SE z value Pr(>|z|)
## (Intercept)      0.0000     0.0000      0.0000      NA       NA
## DistPred         0.2163     0.1398      0.1433    1.51     0.13
## HabOpen.CloseO   0.0162     0.1428      0.1465    0.11     0.91
## BushWoodGrassG   0.1886     0.1909      0.1958    0.96     0.34
## BushWoodGrassW   0.0453     0.1916      0.1966    0.23     0.82
##
## Full model-averaged coefficients (with shrinkage):
##                Estimate Std. Error Adjusted SE z value Pr(>|z|)
## (Intercept)     0.00000    0.00000     0.00000      NA       NA
## DistPred        0.15427    0.15330     0.15563    0.99     0.32
## HabOpen.CloseO  0.00292    0.06089     0.06242    0.05     0.96
## BushWoodGrassG  0.04124    0.11849     0.12026    0.34     0.73
## BushWoodGrassW  0.00990    0.09155     0.09382    0.11     0.92
##
## Relative variable importance:
##                      DistPred BushWoodGrass HabOpen.Close
## Importance:          0.71     0.22          0.18
## N containing models:    2        2             1
```

when you just fit one model and get a regression coefficient for the effect of a predictor variable, the estimate of that regression coefficient is **conditional on the model you used**. As we saw in R Exercise 3A, including or excluding other predictors can have a big effect on the regression coefficient. Model averaging reduces this problem, by averaging the coefficient for each predictor across all the models that contain it. These are weighted averages, and the AIC weights, which sum to one across all of the models, are used as weights. Model-averaged regression coefficients are sometimes called **unconditional regression coefficients**. While the coefficients are less dependent on any single model, they are still dependent on the model set. The estimates presented under "Full model-averaged coefficients (with shrinkage)"address this problem to some degree. If you examine the output, you'll see that these regression coefficients are smaller than in the table labelled "Model-averaged coefficients" ... this is because the averaage includes a zero for every model that does not contain that variable.

There is no getting around the basic point that your model set must include some good models to provide good inferences. A common way to assess this is to include an intercept-only model, to see if your hypotheses

perform better than a null model that contains no effects on the mean value of the dependent variable. It's sort of ironic that we are back to comparison with a null hypothesis, no? This is one reason why I emphasize that the approaches in CWP CH 2 are not as fundamentally different as some authors suggest. As second reason is that model selection using AIC scores often uses a cut-off of $\Delta AIC = 2$ to identify models with scores close enough to the best model that they are not appreciably worse. This value of $\Delta AIC = 2$ equates to a P-value in a one-to-one manner, as described by in

Murtaugh, P.A. (2014). In defense of P values. Ecology, 95, 611-617.

there is a good discussion of this point in the blog post 'To P or not to P'..