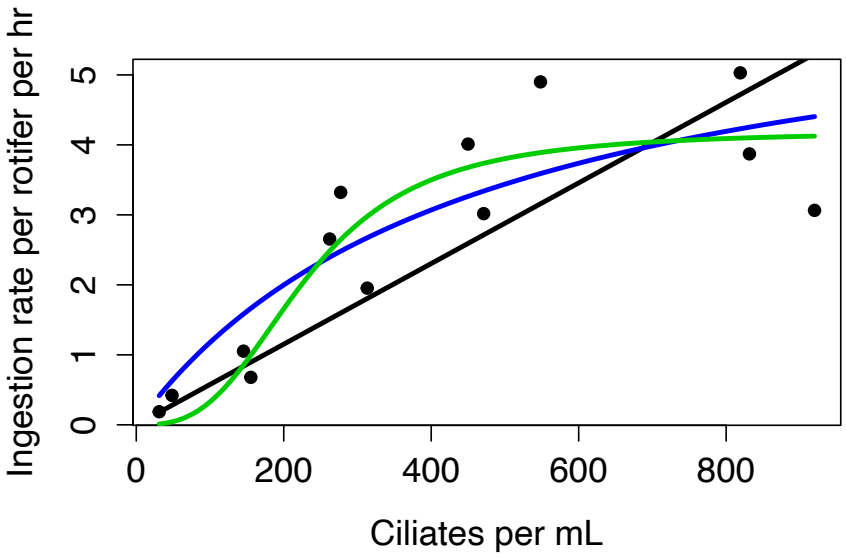
Lecture 15. Model selection with information criteria

A model selection example

This is all pretty abstract so far. Let's look at an example. We'll go back to the different functional responses for rotifers feeding on ciliates. We fit these three models with nls():

```
type1 = nls(ingestion ~ a*ciliates, data = rotifer, start = list(a = 0.01))
type2 = nls(ingestion ~ a*ciliates/(1 + a*h*ciliates), data = rotifer, start = list(a = 0.01, h = 1/4))
type3 = nls(ingestion ~ a*(ciliates^k)/(1 + a*h*(ciliates^k)), data = rotifer, start = list(a = 0.01, h = 1/4, k = 1))
```

The fitted curves look like this:



We would like to compare the relative ability of these models to fit the data. Let's compare AIC for these models. R has a function for AIC, but first let's look at the components of AIC:

```
logLik(type1)

## 'log Lik.' -19.59 (df=2)

logLik(type2)

## 'log Lik.' -14.96 (df=3)

logLik(type3)

## 'log Lik.' -12.91 (df=4)
```

I've used the function logLik() on each of the models. This function returns two numbers: the log-likelihood under the fitted parameters of that model, and the number of parameters, which is called 'df'.

For the three functional response models, the number of parameters (*K*) is 2, 3, and 4 respectively. Notice that the functional response formulas only have 1, 2, and 3 parameters, respectively (e.g., *a* and *h* for the Type 2 curve). What's going on here? R is including the residual variance of the normally distributed error as a parameter that is estimated as part of the model. There seems to be some debate in the literature about why this is, but for now we'll just note that this is included in the parameter calculation.

If we wanted to calculate AIC 'by hand', then we could use the output from logLik, and for the Type 1 model it would be -2*(-19.59) + 2*2 = 43.18. But R has an AIC function that does this for us:

```
AIC(type1)

## [1] 43.18

AIC(type2)

## [1] 35.92

AIC(type3)

## [1] 33.82
```

The AIC() function is part of the base R install, and will work with probably every kind of model you will use. AIC is lowest for the Type 3 functional response (33.82), second-lowest for the Type 2 functional response (35.92), and highest for the Type 1 functional response (43.18). What do these numbers mean? AIC is not useful on an absolute scale, only on a relative scale. The Type 3 model has the lowest AIC, which means it is the best model according to this criterion. Let's compare the best model to the others. The Type 3 model differs from the Type 2 model by 35.92-33.82 = 2.1. This difference is referred to as ΔAIC or Δ_i:

$$\Delta_i = AIC_i - AIC_{min}$$

Where AIC_{min} is the lowest AIC among the set of models being compared (the 'candidate set'), and AIC_i is AIC for model *i*. ΔAIC for the Type 1 model is 43.18-35.92 = 9.35.

ΔAIC compares each model to the best model, on relative terms, but how can we interpret these numbers? The information theory approach is either liberating or frustrating, depending on your point of view. There is no threshold like $p<0.05$ that is used to arbitrarily divide between 'significant' differences and 'non-significant'

differences. However, there are some rules of thumb suggested by the popularizers of AIC:

Δ_i	Level of Empirical Support of Model i
0-2	Substantial
4-7	Considerably less
> 10	Essentially none.

These are suggestions for how we interpret the amount of support for model i, relative to the best model. For example, if the second-best model has a Δ_i of 1, then there is still substantial support for the second-best model, in addition to the best model. While if Δ_i is 10, then there is little support for the second-best model, and the best model is clearly better.

It's not clear where Burnham and Anderson get these recommendations, other than their considerable experience. We'll see some other ways to interpret ΔAIC shortly, but for now let's go back to the function response example. The Type 2 functional response has a ΔAIC of 2.1, relative to the best model. We can interpret this to mean that the Type 3 model is better, but not obviously better. In contrast the Type 1 model has a ΔAIC of 9.35, which means that model has little support from the data.

These numbers were calculated with the standard AIC, but the amount of data used to fit these curves (13 observations) is not huge relative to the number of parameters (1-3). Burnham and Anderson recommend using AICc when $n/K < 40$. For the Type 3 model n/K = 13/3 = 4.3, so the sample size is pretty small relative to the number of parameters. Let's do the same comparisons with AICc:

```
library(MuMIn)
AICc(type1)

## [1] 44.38

AICc(type2)

## [1] 38.59

AICc(type3)

## [1] 38.82
```

There is no function for AICc in basic R, so I've loaded the package 'MuMIn', which we will use for various model selection functions. It looks like with AICc the Type 2 model is now the best model (barely), and ΔAIC is 0.23. So the relative support for the Type 2 and Type 3 models is essentially the same.

AIC weights

Because AIC is a relative metric, not an absolute one, ΔAIC is the fundamental thing we want to know when comparing models. By definition Δ_i is only comparing two models (the best model, and model *i*). Is there a way to take a whole set of models, which we will call the candidate set, and evaluate the relative support for each model, using AIC?

This is typically done using an approximation for the *likelihood of the model*. Specifically,

$$L(m_i|Y) \sim \exp\left(-\frac{1}{2}\Delta_i\right)$$

This is saying the likelihood of model m_i being the best model is proportional to $\exp\left(-\frac{1}{2}\Delta_i\right)$. What is the "likelihood of the model"? This is a somewhat tricky concept, which we will explore more thoroughly when we get to Bayesian methods. For now we'll just say that we're comparing a candidate set of different models, and we want to know the probability that a model is the best model in the set. The formula above is an approximation that allows us to calculate the relative probabilities that different models are the 'best'.

This concept is somewhat confusing, because now we have two different kinds of likelihood:

1. The likelihood of parameter values in some model: $L(\theta|Y)$
2. The likelihood of the model itself: $L(m_i|Y)$

In practice we can standardize the likelihood of the model to get an *Akaike weight*:

$$w_i = \frac{\exp\left(-\frac{1}{2}\Delta_i\right)}{\sum_j^R \exp\left(-\frac{1}{2}\Delta_j\right)}$$

Here we're standardizing the model likelihood by the sum of the model likelihoods for all models in the candidate set, where *R* is the number of models in the set. So the Akaike weight for model *i*, w_i, is the probability that model *i* is the best model, relative to all models in the candidate set. Therefore, *the Akaike weights all us to quantify model selection uncertainty*, i.e. the degree to which we are uncertain which model is the best model according to the information theory approach. We will discuss model selection uncertainty in greater depth next lecture.

Going from AIC, to Δ_i, to the model likelihood, to the Akaike weight seems a bit complex, but in practice is relatively straightforward, and we will be learning to use

a package that automates these steps. Let's calculate Akaike weights for the functional response example, just to clarify what the steps are:

```
#use delta-AIC to calculate the model likelihoods
model.like.1 = exp(-0.5*(AICc(type1) - AICc(type3)))
model.like.2 = exp(-0.5*(AICc(type2) - AICc(type3)))
model.like.3 = exp(-0.5*(AICc(type3) - AICc(type3)))

#sum the model likelihoods for standardization
summed.likes = sum(c(model.like.1, model.like.2, model.like.3))

#calculate the Akaike weights
weight1 = model.like.1/summed.likes
weight2 = model.like.2/summed.likes
weight3 = model.like.3/summed.likes
```

Here I've used AICc. Now let's put the Δ_i and weights in a table for comparison:

Model	Δ_i	w_i
Type 1	5.55	0.03
Type 2	0	0.51
Type 3	0.23	0.46

To sum up the results from this analysis, we can say that the Type 2 model has the highest weight, but it is nearly equally probable that the Type 3 model is the best model. By contrast there is only a 3% chance that the Type 1 model is the best model. This kind of table is the typical result you want to report for the information theory / AIC / Burnham&Anderson protocol for model selection.

Reporting and interpreting model selection results

AIC is a relative metric, useful for selection among models. It doesn't tell you how well a model fits the data in an absolute sense, and it doesn't tell you if your model is even appropriate for the data. Therefore, AIC needs to be combined with other procedures we've been using. Specifically, it is important to plot the data and the fitted model (when possible), and to look at the parameter estimates and their standard errors or confidence intervals. For the functional response models, we would want to report the model selection table, plot the different fitted curves (we already did this), and report the coefficient estimates and standard errors for the different curves. The final step is to think about the meaning/implication of the results, which depends on the goal of the research. For the functional response example, we might say that the consumption rate of the rotifers saturates around 7 ciliates per hour, but that the curve is roughly linear up to about 400 ciliates mL^{-1}, which is a pretty high density. Therefore the functional response may be approximately linear under most conditions. Also, there is some evidence of a

sigmoidal relationship, which would reduce how effective a predator the rotifer is when ciliates are rare, but the data are equivocal on whether this really happens.

AIC and cross-validation

One interesting theoretical result may help make AIC more intuitive. Stone (1977) showed that the ranking of models by AIC is asymptotically equivalent to the ranking of models by leave-one-out cross-validation (LOOCV). Let's think about what that means. We've already covered LOOCV, and we know that it approximates the predictive performance of a model when confronted with new data. If AIC is approximately equal to LOOCV, in terms of the ranking of models, then *AIC is essentially a quick way to estimate the relative predictive performance of models*.

This also sheds some light on what it means when we say that AIC tells us the relative ability of models to approximate the 'true model' that generates the data. According to the cross-validation result, a model that better approximates reality also does a better job of predicting new data.

Let's compare the results of AIC and cross-validation for the functional response example. The ranking of models by AIC and cross-validation is *asymptotically* equivalent. In this case, asymptotically means that these two methods are equivalent under very large sample size, so we may not get identical results under smaller sample sizes. Here's an example of calculating leave-one-out cross-validation with an nls() model:

```
errors = vector()
for (i in 1:nrow(rotifer)) {
  datause = rotifer[-i,]
  mod3 = nls(ingestion ~ a*(ciliates^k)/(1 + a*h*(ciliates^k)), data =
datause, start = list(a = 0.01, h = 1/4, k = 1))
  errors[i] = rotifer$ingestion[i] - predict(mod3, data.frame(ciliates
= rotifer$ciliates[i]))
}
cv3 = sqrt(mean(errors^2))
```

I find that the RMSE for cross-validation is 1.28, 0.94, and 0.87 for the Type 1, Type 2, and Type 3 models, respectively. This approach suggests that the Type 3 model has the best predictive performance, but the difference between the Type 2 and Type 3 models is not large. So these results are similar to what we get from AIC and AICc, in the sense that both models have similar support. It's clear that 13 observations is a small sample size, so we can't put too much stock in any of the comparisons, especially if they yield minor differences.

Model selection with multiple predictors

So far I've presented an example of using AIC to compare different nonlinear curves that represent distinct hypotheses for how nature works. This kind of scenario, where different models represent different hypotheses, is probably the most straightforward model selection problem, particularly when using the information theoretic approach. This is because the relative support for the different models also tells you the relative support for the different hypotheses. It is therefore ironic that AIC seems to be used most commonly in situations when it is difficult to define a small set of models that represent distinct hypotheses. These situations are the subject of this lecture.

To start let's consider a relative simple case: We have a small number of predictors that we want to analyze with a GLM, to see which predictors are most important for explaining the response variable. This kind of scenario is complex for several reasons:

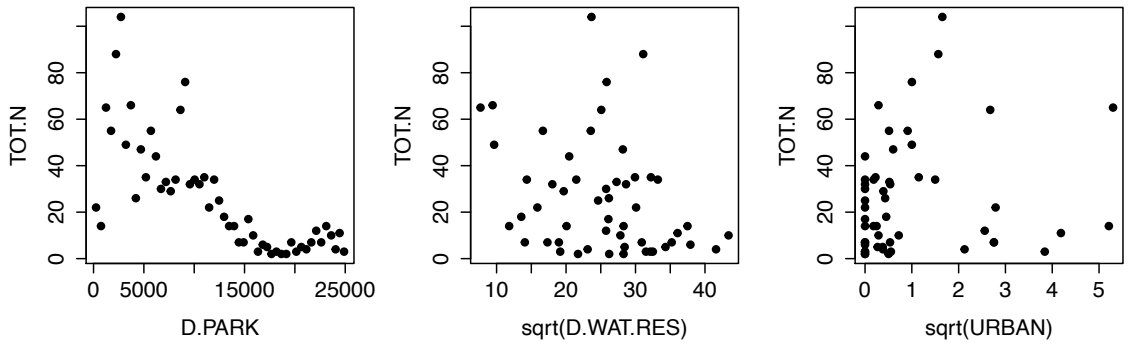
- If you care mostly about testing predictors: there are multiple models that could include/exclude a predictor of interest. Which of these models should we use to draw inferences about that predictor?
- If you care mostly about making predictions: if one model is not overwhelmingly the best model, then how should information from multiple models be used?

I will focus mostly on the first issue now, and the second issue later in the lecture.

As an example, let's look at a dataset of amphibian roadkill observations in Portugal. 52 sites were sampled along a road in southern Portugal (each 'site' is a 500m stretch of road, see map above). Imagine we want to determine what conditions promote roadkills, perhaps to mitigate them in the future. The survey data includes several amphibian variables, and associated environmental variables that could help explain the occurrence of roadkills. For example, roadkills may be more likely if the road crosses a migration route to spawning sites, or roadkills may be more likely during juvenile dispersal. For this exercise let's focus on TOT.N, the total number of roadkills observed during the sampling period; D.PARK, the distance to Serra de São Mamede National Park, which has good amphibian habitat (meters); URBAN, the area of urban habitat surrounding the site (hectares); and D.WAT.RES, the distance to a water reservoir. We want to explore which of these variables are good predictors of the total number of roadkills. Because # of roadkills is a (discrete) count, and sometimes the # of roadkills can be small, it makes sense to model this response variable using a discrete distribution appropriate for count data.

Maybe we hypothesize that roadkill will be greater in areas that have better amphibian habitat, such as areas near parks or near reservoirs. Roadkill may also be higher in areas that are more urbanized, because there is more traffic.

Let's imagine that these three predictors (distance to nearest park, distance to nearest reservoir, amount of urbanized habitat) represent the three hypothesized effects we're interested in testing. The raw data for these relationship look like this:



where I'm plotting number of roadkill observations (TOT.N) vs. the distance to the nearest park, the distance to the nearest reservoir (square root transformed), and the amount of urbanized area in the region (square root transformed). I'll analyze these relationships using a negative binomial distribution, with the standard log link function.

In the homework assignments so far we would approach this problem by making a model that has all three predictors, using that model to plot the fitted effects, and

testing the significance of each predictor using a marginal likelihood ratio test. In this case I think this is a perfectly fine approach. But let's think about how to ask the same kind of questions using an AIC-based approach. This will serve as a comparison of the different approaches, and as a prelude to more complex cases.

There are two different kinds of question we might ask:

1. Which predictors should we include in a model to get the 'best' model for explaining roadkills?
2. What is the relative importance of the different predictors in explaining roadkills?

These are related but distinct questions. The first question is truly a model selection question, i.e. what is the best model. The second question is about the individual predictors. It is a surprisingly tricky question to answer, and it can be useful to apply model selection to answer this question as well. By comparing a number of different models that add/remove the different predictors, we might be able to get a better sense for the importance of a predictor.

It is generally recommended that we think hard in advance about the models we want to compare using AIC. We should make sure that the models are sensible for the data at hand, and that the models in the candidate set allow us to answer specific questions that we care about. For the roadkills question, we want to know whether a given predictor is important for explaining roadkill variation, and the relative importance of different predictors. How can we answer these questions? I want to emphasize that there is no single approach that is universally agreed upon, and any analysis of this type involves a lot of judgement calls. Rather than give you a protocol to follow, I'm going to look at a couple ways one might approach this question, and discuss whether they seem satisfactory or not.

Comparing univariate models

If we want to compare the importance of three predictors, and estimate their effects, maybe we should just fit three models with a single predictor each:

```
model.P = glm.nb(TOT.N ~ D.PARK, data = roadkills)
model.U = glm.nb(TOT.N ~ sqrt(URBAN), data = roadkills)
model.W = glm.nb(TOT.N ~ sqrt(D.WAT.RES), data = roadkills)
```

To make an AIC model selection table, we can use a nice function from the package 'MuMIn':

```
library(MuMIn)
all.models = list(model.P, model.U, model.W)
aic.table = model.sel(all.models)
aic.table
```

```
## Model selection table
##   (Int) D.PAR      sqr(URB) sqr(D.WAT.RES) family    init.theta df logLik
## 1 4.411 -0.0001161                       NB(3.681) 3.68       3  -193.5
## 3 4.179                    -0.0383        NB(1.294) 1.29       3  -219.2
## 2 3.146            0.1022                 NB(1.203) 1.2        3  -221.4
##   AICc  delta weight
## 1 393.6  0.00 1
## 3 445.0 51.36 0
## 2 449.2 55.63 0
```

The function model.sel will compute AICc for each model, Δ_i for each model (column 'delta'), and the Akaike weight for each model (column 'weight'). It also report the parameter estimates for each model, the log-likelihood, the number of parameters in the model, etc. And it sorts them so that the lowest AIC value is at the top.

What do we learn from this? The model with D.PARK has a much lower AICc than the other two models (difference of ~50), and the Akaike weight for that model is ~1. So if we wanted to compare these three predictors as alternative hypotheses, clearly distance to parks is the best predictor.

However, this doesn't seem like a great approach. Even if D.PARK is the best predictor, the others could be important for explaining or predicting the number of roadkills. It is likely that the best model, according to AIC, would have multiple predictors in many situations.

Making marginal comparisons

When we analyzed this data in the homework assignment, we created one model with all the predictors, and then looked at the effect of removing each predictor individually. This essentially asks whether a predictor is important, when the other predictors are already in the model. We could also do this with the AIC model selection approach:

```
model.PU = glm.nb(TOT.N ~ D.PARK + sqrt(URBAN), data = roadkills)
model.PW = glm.nb(TOT.N ~ D.PARK + sqrt(D.WAT.RES), data = roadkills)
model.UW = glm.nb(TOT.N ~  sqrt(URBAN) + sqrt(D.WAT.RES), data = roadkills)
model.PUW = glm.nb(TOT.N ~ D.PARK + sqrt(URBAN) + sqrt(D.WAT.RES), data =
roadkills)
```

Here I've made four models: the full model (model.PUW), and one model that removes each predictor. Now make a model selection table:

```
all.models = list(model.PU, model.PW, model.UW, model.PUW)
```

```
aic.table = model.sel(all.models)
aic.table
```

```
Model selection table
  (Int) D.PAR      sqr(URB) sqr(D.WAT.RES) family     init.theta df logLik   AICc  delta weight
2 4.031 -0.0001299          0.02155       NB(3.9665) 3.97       4  -191.885 392.6  0.00 0.609
4 3.945 -0.0001310 0.05456   0.02342       NB(3.9694) 3.97       5  -191.518 394.3  1.72 0.258
1 4.378 -0.0001162 0.03414                 NB(3.6726) 3.67       4  -193.412 395.7  3.05 0.132
3 4.095            0.04291  -0.03669      NB(1.2974) 1.3        4  -219.151 447.2 54.53 0.000
```

Let's think about these results. The 'best' model is the model with D.PARK and D.WAT.RES, but not URBAN. And this model has a weight of 0.61. The second-best model is the full model that also has URBAN as a predictor. Δ_i for this model is 1.72, and its weight is 0.26. In combination this suggests that distance to park and distance to reservoir are both important predictors of roadkills, because they are in the top two models, which have a combined weight of 0.867. This also suggests that although the best model does not have URBAN as a predictor, including URBAN does not make the model much worse. I will return to what this means later.

I don't find these results totally satistfying yet. To get a better sense for the importance of D.PARK and D.WAT.RES, we can see what happens when they are removed from the full model. The full model has a weight of 0.258, while the model with D.PARK removed has a weight of ~0, and the AIC difference between these models is ~52. Clearly D.PARK is an important predictor. The model with D.WAT.RES removed has a weight of 0.132. If we compare this to the full model, the ratio is 0.258/0.132 = 1.95. This is called the *evidence ratio* for these two models. The evidence ratio is telling us that the full model is about twice as likely to be the best model, compared to the model with D.WAT.RES removed. So this suggest that there is some evidence that reservoirs are important, but it's not really strong evidence.

This approach, comparing a full model and models with predictors removed, seems defensible and informative. However, there are questions we can't answer with this candidate set of models, questions which we might be interested in. For example, the model with distance to parks and distance to reservoirs is the best model, but how does this compare to a model with just distance to parks? Maybe this simple model is actually the best.

<u>Comparing all possible models</u>

Let's imagine that I want a more comprehensive view of the role of these predictors. The most extreme approach is to construct *all possible models containing these predictors*. There are functions that can automate this process, but for now I'll do it manually:

```
model.0 = glm.nb(TOT.N ~ 1, data = roadkills)
model.P = glm.nb(TOT.N ~ D.PARK, data = roadkills)
model.U = glm.nb(TOT.N ~ sqrt(URBAN), data = roadkills)
model.W = glm.nb(TOT.N ~ sqrt(D.WAT.RES), data = roadkills)
model.PU = glm.nb(TOT.N ~ D.PARK + sqrt(URBAN), data = roadkills)
model.PW = glm.nb(TOT.N ~ D.PARK + sqrt(D.WAT.RES), data = roadkills)
model.UW = glm.nb(TOT.N ~  sqrt(URBAN) + sqrt(D.WAT.RES), data = roadkills)
model.PUW = glm.nb(TOT.N ~ D.PARK + sqrt(URBAN) + sqrt(D.WAT.RES), data =
roadkills)
```

I've created 8 models. These include every possible combination of each predictor being present/absent from the model. Note that I have not included interactions between the predictors. That would increase the number of models greatly; here I'm keeping it simple, but in some cases you may want to include all possible two-way interactions, for example, or perhaps only those interactions that seem biologically plausible.

Let's make a model selection table with all possible models:

```
all.models = list(model.0, model.P, model.U, model.W, model.PU,
model.PW, model.UW, model.PUW)

aic.table = model.sel(all.models)
aic.table
```

```
Model selection table
  (Int) D.PAR      sqr(URB) sqr(D.WAT.RES) family     init.theta df logLik   AICc  delta weight
6 4.031 -0.0001299           0.02155       NB(3.9665) 3.97       4  -191.885 392.6  0.00 0.443
2 4.411 -0.0001161                         NB(3.681)  3.68       3  -193.546 393.6  0.97 0.273
8 3.945 -0.0001310 0.05456   0.02342       NB(3.9694) 3.97       5  -191.518 394.3  1.72 0.188
5 4.378 -0.0001162 0.03414                 NB(3.6726) 3.67       4  -193.412 395.7  3.05 0.096
4 4.179                     -0.03830       NB(1.294)  1.29       3  -219.226 445.0 52.33 0.000
7 4.095            0.04291  -0.03669       NB(1.2974) 1.3        4  -219.151 447.2 54.53 0.000
1 3.254                                    NB(1.1837) 1.18       2  -221.832 447.9 55.29 0.000
3 3.146            0.10220                 NB(1.2027) 1.2        3  -221.361 449.2 56.60 0.000
```

What do we learn from these results? The model with D.PAR and D.WAT.RES is again the best model. But now the second-best model has only D.PAR as the predictor, while the third-best model is the model with all predictors. All three of these models get decent support from the data. It's clear that distance to parks is essential for constructing a good model, while the distance to reservoirs has some support but is maybe less essential. So the conclusions from the all-possible-models approach is similar to the the conclusions from the marginal-comparisons approach.

It can be challenging to compare many models in terms of Δ_i and model weights. One way to summarize the information is in terms of *variable importance*, which is

calculated by *summing the weights of all the models that include a variable*. This can be automated with the function importance():

```
importance(all.models)
```

```
##                       D.PARK sqrt(D.WAT.RES) sqrt(URBAN)
## Importance:           1.00   0.63            0.28
## N containing models:    4      4               4
```

The importance for D.PARK is 1; only models with this variable have any support. The importance for D.WAT.RES is 0.63, which suggests that this variable is of intermediate importance but not absolutely necessary for making a good model. The importance for URBAN is 0.28, which is pretty weak support.

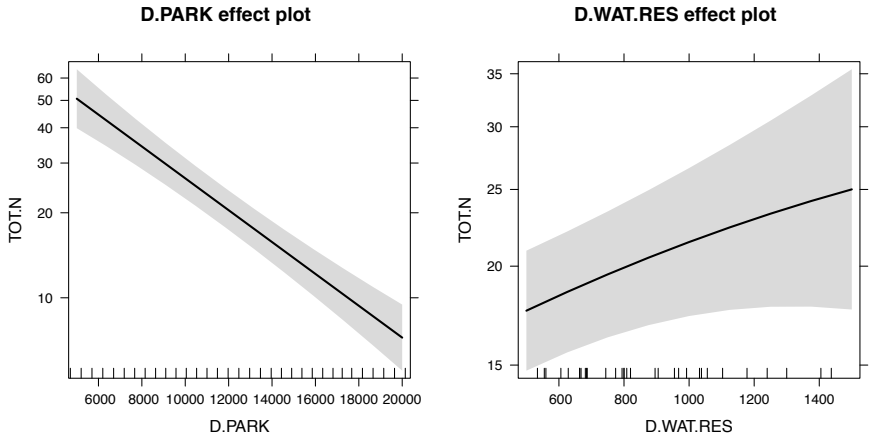
Calculating variable importance is something that should be done with caution. The results will depend on the candidate set of models, and if one variable appears in more models than another variable, it will tend to have higher importance. By constructing all possible models, we have a situation in which each predictor is present in 4 models and absent from 4 models, so this is a good situation in which to compare variable importance. It is also the case that *variable importances are relative to which variables are being compared*. If one predictor were not in any of the models, the results could change greatly.

What is the best approach for constructing models with multiple predictors?

Is it better to construct all possible models, or just a subset of models to make targeted comparisons, as I did previously? This depends on the analysis, and I have seen practitioners use many different approaches. Burnham & Anderson try to avoid comparing all possible models, as it seems more like data dredging and less like testing hypotheses. But depending on your goals, it has the advantage of giving a more complete picture of what's going on. Rather than give a specific recommendation, I'm going to continue with more examples, and ultimately you will get a better feel for what the options are and when you might want to use them.

Model selection can help us get a handle on the importance of different predictors, *but by itself it is incomplete*. To turn these results into a formal analysis we need to combine these results with plots of fitted effects, or a table of coefficient estimates and their standard errors, which would help us understand the importance of predictors in terms of effect size. If there is a clear 'best model', i.e. one with a model weight > 0.9, then we can just report the results from that model. If there are multiple models that have support then reporting the fitted effects is more complicated. If the fitted effects differ between competing models, we may want to report the coefficients from multiple models. In the case of the roadkill data, the

slope estimates for the three predictors do not vary greatly among the top models. In this case I would probably plot the results from the top model, because it includes the two predictors (D.PARK, D.WAT.RES) that have support from the data:



We should also assess how well the model fits the data in general (e.g. pseudo-R^2).

AIC vs. null hypothesis testing

For this example, i.e. comparing the importance of a small number of predictors, it seems reasonable to use either the information criterion approach or the null hypothesis testing approach. They also give similar results. This is what Anova looks like on the full model:

```
## Analysis of Deviance Table (Type II tests)
##
## Response: TOT.N
##                 LR Chisq Df Pr(>Chisq)
## D.PARK              96.1  1     <2e-16 ***
## sqrt(URBAN)          0.7  1      0.391
## sqrt(D.WAT.RES)      3.9  1      0.048 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From this we would reach essentially the same conclusions as from the AIC approach: distance to park is very important, distance to reservoir has some support but is more equivocal, and urbanization has little support.

The fact that we get similar results is reassuring, and I would support using either approach in this case. The marginal null hypothesis tests are a bit easier to implement, but some people may prefer AIC for philosophical reasons.

Effect of a useless predictor on AIC

When using ΔAIC to select among models, there is an important asymmetry that is often unacknowledged, having to do with the difference between removing an important predictor and adding an unimportant predictor.

Removing an important predictor from a model can have a large effect on AIC. For example, in the roadkill models compared above, dropping the predictor for distance to parks causes AIC to increase by about 50. This makes sense, because adding a predictor that explains a lot of variation will greatly increase the model likelihood, and thereby decrease AIC.

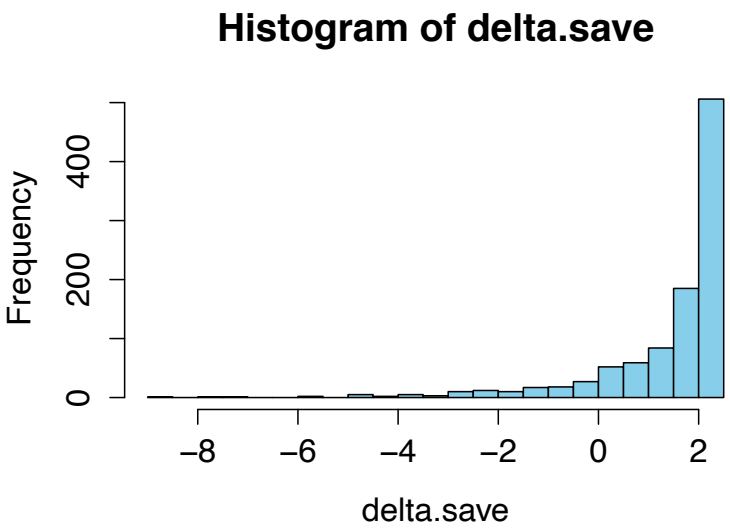
At the other extreme is a predictor that has no explanatory power. What happens when such a predictor is added/removed from a model? Let's do a simulation to find out. I'm going to take the same roadkills dataset, and make a new column that is just random numbers. On average this column should not be correlated with the response variable (roadkills), but by chance this column could show a spurious correlation with the response. I'll make this fake predictor 1000 times, each time comparing the AIC of the best model (D.PARK + D.WAT.RES) to a model that also includes the fake predictor:

```
delta.save = vector()
for (i in 1:1000) {

  roadkills$fake.predictor = rnorm(nrow(roadkills))
  model.fake = glm.nb(TOT.N ~ D.PARK + sqrt(D.WAT.RES) + fake.predictor, data
= roadkills)
  delta.save[i] = AICc(model.fake) - AICc(model.PW)

  }

hist(delta.save, col = 'skyblue', breaks = 20)
```



The histogram shows the distribution of ΔAICc = (AICc(best model) – AICc(model with fake predictor)). In most cases ΔAICc is about 2. This makes sense, because the formula for AIC,

$$\text{AIC} = -2 * \log\left(L\left(\hat{\theta}\middle|Y\right)\right) + 2K$$

shows that if a predictor doesn't increase the likelihood very much (the first term), then AIC should increase by about 2*(number of additional parameters). In this case there is one additional parameter, so AIC should increase by about 2.

You can also see that sometimes the fake predictor cases AICc to decline, i.e. ΔAICc < 0. These are cases where the fake predictor has a spurious relationship with #roadkills. The important message here is that *an important predictor can cause a large change in AIC, but a useless predictor can only cause AIC to increase by about 2*. This is an important asymmetry to keep in mind when using AIC to compare models. In the roadkills example, adding URBAN to the best model increases AIC by about 1.7. According to Burnham & Anderson's guidelines, a difference of 1.7 is not enough to say that the model with URBAN has no support from the data. But at the same time it is clear that adding URBAN does not improve the model, rather it makes it worse. Therefore, when evaluating the importance of predictor it is necessary to think about whether adding a predictor is really helping or not.

This asymmetry for AIC makes sense, if we think about it. It is possible for the data to provide strong evidence that a predictor is important, but it is much more difficult to have strong evidence that a predictor adds zero explanatory power. Rather, AIC will tell us that including or excluding a predictor may have little effect on the ability of the model to approximate reality. This may be because a predictor is truly zero, or because a predictor has a real effect that is very small.