Model selection and multimodel inference using the `AICcmodavg` package

Marc J. Mazerolle[*]

20 March 2023

Abstract

The `AICcmodavg` package implements model selection and multimodel inference for a wide range of model types. This vignette outlines the first steps to use the package and also presents the main functions. The package also offers utility functions for diagnostics and enhancements to specific classes of models that estimate demographic parameters and vital rates in populations of unmarked animals (Fiske and Chandler, 2011).

1 Introduction

The publication of Burnham and Anderson (1998) and an expanded second edition of the book four years later (Burnham and Anderson, 2002) initiated a shift in ecology from traditional null-hypothesis statistical testing to the adoption of information-theoretic approaches for model selection and inference. This movement also echoed a broader fundamental change of focus in statistical inference. Whereas many statistical approaches have traditionnally centered on null-hypothesis statistical testing and P-values, emphasis has moved to estimating parameters and measures of uncertainty (Goodman, 1999; Nuzzo, 2014; Wasserstein et al., 2019; Calin-Jageman and Cumming, 2019; Anderson, 2019).

The `AICcmodavg` package implements model selection and multimodel inference based on different information criteria, including AIC, AIC_c, $QAIC$, $QAIC_c$, and BIC (Akaike, 1973; Sugiura, 1978; Burnham and Anderson, 2002; Schwarz, 1978). Before starting analyses, I suggest considering the ten following guidelines:

1. **Carefully construct your candidate model set.** Each model should represent a specific (interesting) hypothesis to test. Thought needs to be put in the models that are relevant for the hypotheses and data at hand.
2. **Keep your candidate model set short.** The number of models should generally be less than the number of data points (Burnham and Anderson, 2002).

[*] Département des sciences du bois et de la forêt, Université Laval, Québec, Canada

3. **Check model fit.** Use the global model (i.e., the model from which all other models can be derived) to assess model fit and ensure that model assumptions are met. If none of your models fit the data well, information criteria will only indicate the most parsimonious of the poor models.

4. **Avoid data dredging.** Data dredging or data snooping consists in running analyses to find effects in your model set and then building the candidate model set based on this information. This is ill-advised and you should avoid such a procedure. You should specify the candidate model set based on your hypotheses, and then do model selection based on this model set.

5. **Avoid overfitting models.** You should not estimate too many parameters for the number of observations available in the sample. Running a model much too complex for the data available can lead to spurious results.

6. **Watch out for missing values.** Values that are missing only for certain explanatory variables change the data set and sample size, depending on which variable is included in any given model. You should deal with missing values before analysis, either by deleting certain observations or using missing data imputation (Gelman and Hill, 2007).

7. **Use the same response variable for all models of the candidate model set.** It is inappropriate to run some models with a transformed response variable and others with the untransformed variable. A workaround is to use a different link function for some models (McCullagh and Nelder, 1989).

8. **When dealing with models with overdispersion, use the same value of $\hat{c}$ for all models in the candidate model set.** Overdispersion occurs in certain models that use binomial or Poisson distributions and results from the variance in the data exceeding that allowed by the distribution. One way to diagnose the presence of overdispersion is to estimate a variance inflation factor ($\hat{c}$) from the global model. Note that functions `c_hat( )`, `mb.gof.test( )`, and `Nmix.gof.test( )` estimate $\hat{c}$ for specific model types.

9. **Avoid mixing the information-theoretic approach and notions of statistical significance (i.e., *P* values).** Information criteria and P-values do not mix (Burnham and Anderson, 2002). Instead, you should provide estimates and a measure of their precision such as unconditional standard errors or confidence intervals).

10. **Determining the ranking of the models is just the first step.** When the top-ranking model has most of the support (e.g., Akaike weights > 0.9), it can be appropriate to base inference on this single most parsimonious model. However, when many models rank highly, one should model-average effect sizes for the parameters with most support across the entire set of models. This is the underlying idea behind multimodel inference which consists in making inference based on the whole set of candidate models.

After this preamble, we can start with an example using various functions of the `AICcmodavg` package.

2 Getting started

In this section, we will walk through the steps to building the models as well as conducting model selection and multimodel inference with an example data set. Here, we will use the `dry.frog` data set from Mazerolle (2006). The data feature mass lost by green frogs (*Lithobates clamitans*) after spending two hours on one of three substrates that are encountered in some landscape types (for additional details, check Mazerolle and Desrochers 2005). The response variable is the mass lost (`Mass_lost`) and we are interested in testing difference among substrate types. To simplify the example, we will only consider main effects, but note that you should consider interactions whenever relevant (Mazerolle and Desrochers 2005 include interaction terms in the analysis).

2.1 Importing data

Usually, importing a typical dataset in `R` will involve using `read.table( )` or one of its variations (e.g., `read.csv( )`, `read.delim( )`). In our case, the data set is already included in the `AICcmodavg` package, so we will load it directly:

```
> library(AICcmodavg)
> data(dry.frog)
```

For this example, we'll only be using the first seven columns of the data set:

```
> ##extract only first 7 columns
> frog <- dry.frog[, 1:7]
> ##first lines
> head(frog)
  Individual Species Shade  SVL Substrate Initial_mass Mass_lost
1          1   Racla     0 7.27      SOIL         38.5       8.3
2          2   Racla     0 7.00  SPHAGNUM         31.0       3.6
3          3   Racla     0 6.83      PEAT         23.6       4.7
4          4   Racla     0 7.26      PEAT         37.4       7.0
5          5   Racla     0 7.43      SOIL         44.4       7.7
6          6   Racla     0 5.75  SPHAGNUM         16.4       1.6
> ##structure of data frame
> str(frog)
'data.frame':        121 obs. of  7 variables:
 $ Individual  : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Species     : Factor w/ 1 level "Racla": 1 1 1 1 1 1 1 1 1 1 ...
 $ Shade       : int  0 0 0 0 0 0 0 0 0 0 ...
 $ SVL         : num  7.27 7 6.83 7.26 7.43 5.75 7.66 6.42 7.64 6.57 ...
 $ Substrate   : Factor w/ 3 levels "PEAT","SOIL",..: 2 3 1 1 2 3 1 2 3 2 ...
 $ Initial_mass: num  38.5 31 23.6 37.4 44.4 16.4 39.8 25.9 35.6 29 ...
 $ Mass_lost   : num  8.3 3.6 4.7 7 7.7 1.6 6.4 5.9 2.8 3.4 ...
```

Note that `Substrate` is a factor with three levels. Using the default treatment contrast coding in `R`, the variable has been recoded automatically with two indicator (dummy) variables.

It's also a good idea to check for missing values:

```
> any(is.na(frog))
```

```
[1] FALSE
```

In this case, there are no missing values and we won't have to worry about some observations being excluded in certain models.

2.2 Specifying candidate models based on hypotheses

We are interested in testing the effect of substrate type on the mass lost by frogs, but a number of potential other variables could also influence the mass lost, namely, the initial mass of individuals (linear and quadratic effects) and the presence of shade (shade vs no shade). Given that mass loss is numeric, we will consider a multiple regression model using the normal distribution. We specify eight candidate models to test our hypotheses. Each parameter appears in four models, which will be a useful condition for multimodel inference (see Inference on β estimates below). We also included a null model to quantify the support in favor of models relative to the null model:

1. Null model
Biological hypothesis: Mass lost by frogs is constant.

$$\hat{Y}_i = \beta_0$$

2. Shade model
Biological hypothesis: Mass lost by frogs varies with shade.

$$\hat{Y}_i = \beta_0 + \beta_{Shade} * Shade_i$$

3. Substrate model
Biological hypothesis: Mass lost by frogs varies with substrate type.

$$\hat{Y}_i = \beta_0 + \beta_{SubstrateSOIL} * SubstrateSOIL_i + \beta_{SubstratePEAT} * SubstratePEAT_i$$

4. Shade and substrate model
Biological hypothesis: Mass lost by frogs varies with shade and substrate type.

$$\hat{Y}_i = \beta_0 + \beta_{Shade} * Shade_i + \beta_{SubstrateSOIL} * SubstrateSOIL_i + \beta_{SubstratePEAT} * SubstratePEAT_i$$

5. Null model with mass
Biological hypothesis: Mass lost by frogs varies with frog size.

$$\hat{Y}_i = \beta_0 + \beta_{Initial\_mass} * Initial\_mass_i + \beta_{Initial\_mass2} * Initial\_mass2_i$$

6. Shade model with mass
Biological hypothesis: Mass lost by frogs varies with frog size and shade.

$$\begin{aligned}\hat{Y_i} =&\beta_0 + \beta_{Initial\_mass} * Initial\_mass_i + \beta_{Initial\_mass2} * Initial\_mass2_i + \\ &\beta_{Shade} * Shade_i\end{aligned}$$

7. Substrate model with mass
Biological hypothesis: Mass lost by frogs varies with frog size and substrate type.

$$\begin{aligned}\hat{Y_i} =&\beta_0 + \beta_{Initial\_mass} * Initial\_mass_i + \beta_{Initial\_mass2} * Initial\_mass2_i + \\ &\beta_{SubstrateSOIL} * SubstrateSOIL_i + \beta_{SubstratePEAT} * SubstratePEAT_i\end{aligned}$$

8. Shade and substrate model with mass
Biological hypothesis: Mass lost by frogs varies with frog size, shade, and substrate type.

$$\begin{aligned}\hat{Y_i} =&\beta_0 + \beta_{Initial\_mass} * Initial\_mass_i + \beta_{Initial\_mass2} * Initial\_mass2_i + \\ &\beta_{Shade} * Shade_i + \beta_{SubstrateSOIL} * SubstrateSOIL_i + \beta_{SubstratePEAT} * SubstratePEAT_i\end{aligned}$$

2.3 Formating data

Some of our hypotheses involve linear and quadratic effects of initial mass. To reduce correlations between the two variables, we will center initial mass by subtracting the mean of the variable from each value:

```
> ##center initial mass
> frog$InitMass_cent <- frog$Initial_mass - mean(frog$Initial_mass)
```

We can then square the centered variable:

```
> frog$InitMass2 <- frog$InitMass_cent^2
```

2.4 Checking global model fit

With these eight candidate models specified, we can now check the diagnostics of the global model.

```
> ##run global model
> global <- lm(Mass_lost ~ InitMass_cent + InitMass2 + Substrate + Shade,
               data = frog)
> par(mfrow = c(2, 2))
> plot(global)
```

The assumption of homoscedasticity does not seem to be met with the raw response variable, as the variance increases with the mean (Fig. 1). To circumvent this issue, we will apply a log transformation to the response variable:

```
> frog$logMass_lost <- log(frog$Mass_lost + 1) #adding 1 due to presence of 0's
```

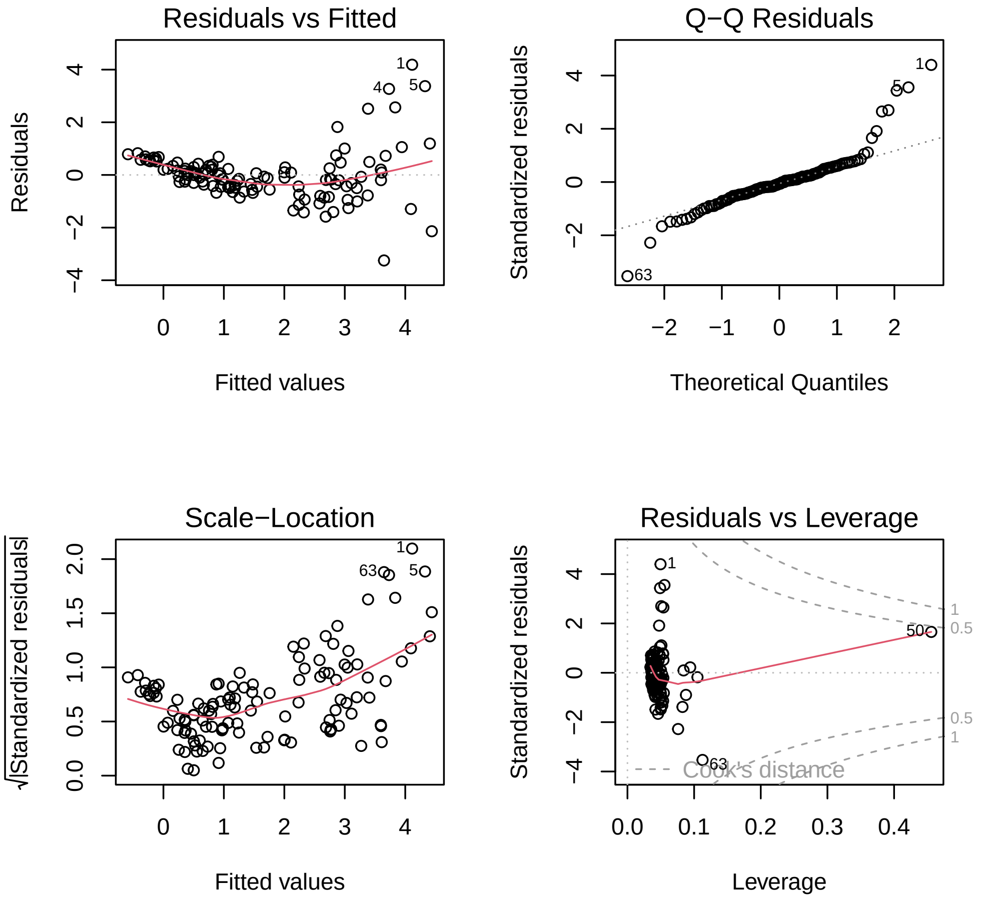

Figure 1: Assessment of model assumptions using residuals and fitted values from the global model based on mass lost (g) by green frogs (*Lithobates clamitans*) exposed to different conditions.

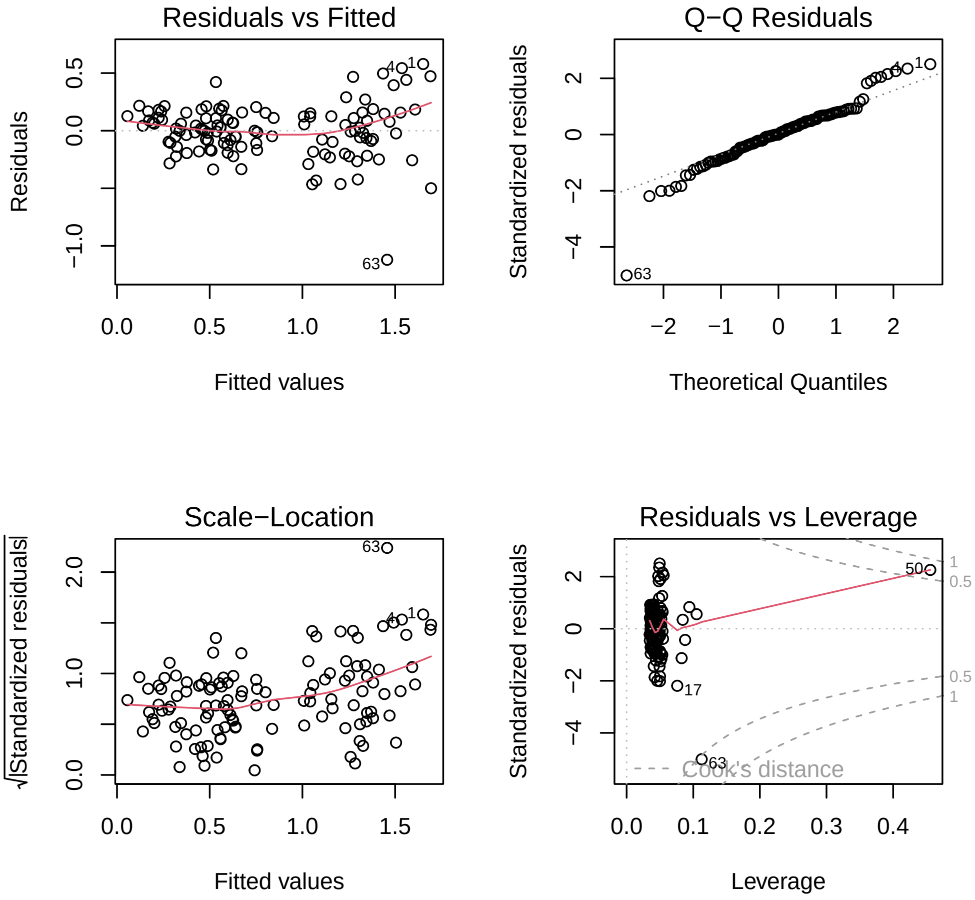

Figure 2: Assessment of model assumptions using residuals and fitted values from the global model based on the log of the mass lost (g) by green frogs (*Lithobates clamitans*) exposed to different conditions.

```
> ##run global model
> global.log <- lm(logMass_lost ~ InitMass_cent + InitMass2 + Substrate + Shade,
                   data = frog)
> par(mfrow = c(2, 2))
> plot(global.log)
```

The log transformation generally homogenized the variance and most residuals follow a normal distribution, except for a few outliers (Fig. 2). Thus, we will proceed with the analysis using the log transformation on all candidate models.

2.5 Running candidate models and saving in list

We fit the 8 models in the candidate set:

```
> m.null <- lm(logMass_lost ~ 1,
               data = frog)
```

```
> m.shade <- lm(logMass_lost ~ Shade,
                data = frog)
> m.substrate <- lm(logMass_lost ~ Substrate,
                    data = frog)
> m.shade.substrate <- lm(logMass_lost ~ Shade + Substrate,
                          data = frog)
> m.null.mass <- lm(logMass_lost ~ InitMass_cent + InitMass2,
                    data = frog)
> m.shade.mass <- lm(logMass_lost ~ InitMass_cent + InitMass2 + Shade,
                     data = frog)
> m.substrate.mass <- lm(logMass_lost ~ InitMass_cent + InitMass2 + Substrate,
                         data = frog)
> m.global.mass <- global.log
```

Most functions for model selection and multimodel inference in the AICcmodavg package require that the output of the candidate models be stored in a single list. Although the functions will add generic names to each model automatically if none are supplied, it is good practice to provide meaningful and succinct names for each model. This will help in the interpretation of the model selection tables. Model names can be entered as a character string using the modnames argument or directly as a named list.

Here are the model outputs stored in a list with names assigned to each element:

```
> ##store models in named list
> Cand.models <- list("null" = m.null, "shade" = m.shade,
                      "substrate" = m.substrate,
                      "shade + substrate" = m.shade.substrate,
                      "mass" = m.null.mass, "mass + shade" = m.shade.mass,
                      "mass + substrate" = m.substrate.mass,
                      "global" = m.global.mass)
```

2.6 Doing model selection

We are now ready to build the model selection table with aictab:

```
> selectionTable <- aictab(cand.set = Cand.models)
> selectionTable

Model selection based on AICc:

                  K   AICc Delta_AICc AICcWt Cum.Wt     LL
global            7   4.07       0.00      1      1   5.46
mass + substrate  6  26.82      22.75      0      1  -7.04
mass + shade      5  32.32      28.25      0      1 -10.90
mass              4  49.44      45.37      0      1 -20.55
shade + substrate 5 172.31     168.24      0      1 -80.89
substrate         4 176.31     172.24      0      1 -83.98
shade             3 180.21     176.14      0      1 -87.00
null              2 183.58     179.51      0      1 -89.74
```

We note that the global model has all the support. By default, AIC_c is used in the model selection and multimodel inference functions, but AIC can be selected with the second.ord = FALSE argument:

```
> aictab(Cand.models, second.ord = FALSE)
Model selection based on AIC:

                  K    AIC Delta_AIC AICWt Cum.Wt     LL
global            7   3.08      0.00     1      1   5.46
mass + substrate  6  26.08     23.00     0      1  -7.04
mass + shade      5  31.80     28.72     0      1 -10.90
mass              4  49.10     46.02     0      1 -20.55
shade + substrate 5 171.79    168.71     0      1 -80.89
substrate         4 175.96    172.88     0      1 -83.98
shade             3 180.00    176.92     0      1 -87.00
null              2 183.48    180.40     0      1 -89.74
```

For those familiar with LaTeX(Lamport, 1994; Mittelbach and Goossens, 2004), note that most functions in `AICcmodavg` can export result tables in LaTeX format using `xtable( )` methods from the `xtable` package (Dahl, 2014). For example, the following code will produce Table 1:

```
> library(xtable)
> print(xtable(selectionTable, caption = "Model selection table on frog mass lost.",
               label = "tab:selection"),
        include.rownames = FALSE, caption.placement = "top")
```

Table 1: Model selection table on frog mass lost.

Model	K	AICc	Delta AICc	AICc weight	log-Likelihood
global	7	4.07	0.00	1.00	5.46
mass + substrate	6	26.82	22.75	0.00	-7.04
mass + shade	5	32.32	28.25	0.00	-10.90
mass	4	49.44	45.37	0.00	-20.55
shade + substrate	5	172.31	168.24	0.00	-80.89
substrate	4	176.31	172.24	0.00	-83.98
shade	3	180.21	176.14	0.00	-87.00
null	2	183.58	179.51	0.00	-89.74

We can provide complementary information to assist the interpretation of the model selection table. For instance, we can compute the 95% confidence set of models (Burnham and Anderson, 2002):

```
> ##confidence set of models
> confset(cand.set = Cand.models)
Confidence set for the best model

Method:          raw sum of model probabilities

95% confidence set:
       K AICc Delta_AICc AICcWt
global 7 4.07          0      1

Model probabilities sum to 1
```

Evidence ratios are also useful to quantify the amount of support in favor of a model relative to a competing model (Burnham and Anderson, 2002). Function evidence( ) takes a model selection table as argument:

```
> ##evidence ratios
> evidence(aic.table = selectionTable)
Evidence ratio between models 'global' and 'mass + substrate':
87087.77
```

Here, we see that the global model is 87088 times more parsimonious that the substrate model. It is also possible to compare two arbitrary models by using their names in the model.high and model.low arguments:

```
> ##compare "substrate" vs "shade"
> evidence(selectionTable, model.high = "substrate",
           model.low = "shade")
Evidence ratio between models 'substrate' and 'shade':
7.04
```

We conclude that the substrate model is 7 times more parsimonious than the shade model. Another useful comparison is between the top-ranked model and the null model:

```
> evidence(selectionTable, model.high = "global",
           model.low = "null")
Evidence ratio between models 'global' and 'null':
9.572308e+38
```

Because the top-ranked model has all the support, we could interpret the results of the model using confidence intervals:

```
> confint(m.global.mass)
                          2.5 %        97.5 %
(Intercept)        1.0119220187  1.2071260941
InitMass_cent      0.0292360282  0.0369301094
InitMass2         -0.0007653636 -0.0003673069
SubstrateSOIL     -0.0032563220  0.2091385826
SubstrateSPHAGNUM -0.3112314181 -0.0994731113
Shade             -0.3081777571 -0.1366728765
```

We conclude that there is a quadratic effect of initial mass on frog mass loss, that mass loss is lower in the presence of shade, and that mass loss is lower on *Sphagnum* moss (living vegetation) than on peat. However, model support will often be shared by several models (i.e., top-ranked model having $< 90\%$ of the support). In such cases, we should conduct multimodel inference.

2.7 Making multimodel inference

Four main functions are used for multimodel inference in the AICcmodavg package: modavg( ), modavgShrink( ), modavgPred( ), and modavgEffect( ). Two of these functions focus on making inferences on β estimates (modavg( ), modavgShrink( )) and the two others work on model predictions (modavgPred( ), modavgEffect( )). These functions are presented below.

2.7.1 Inference on β estimates

Two functions are available to compute model-averaged estimates of β parameters. Function `modavg( )` implements the natural average. This method consists in using exclusively the models that include the parameter of interest, recalculating the ΔAIC and Akaike weights, and computing a weighted average of the estimates (Burnham and Anderson, 2002, p. 152). We can compute the natural average of the effect of shade (β_{Shade}) on the loss of frog mass:

```
> modavg(cand.set = Cand.models, parm = "Shade")
Multimodel inference on "Shade" based on AICc

AICc table used to obtain model-averaged estimate:

                  K   AICc Delta_AICc AICcWt Estimate   SE
shade             3 180.21     176.14      0    -0.21 0.09
shade + substrate 5 172.31     168.24      0    -0.22 0.09
mass + shade      5  32.32      28.25      0    -0.22 0.05
global            7   4.07       0.00      1    -0.22 0.04

Model-averaged estimate: -0.22
Unconditional SE: 0.04
95% Unconditional confidence interval: -0.31, -0.14
```

Note that the table only features the models that include shade as an explanatory variable. We conclude that frogs loose less mass in the shade than out of the shade lost ($\hat{\bar{\beta}}_{Shade} = -0.22$, 95% CI: $[-0.31, -0.14]$).

Similarly, we can request a model-averaged estimate for factor levels, keeping in mind that only certain contrast have been estimated in the model (i.e., there are three levels, but only two contrasts). Note that the parameter must be specified with the same label as in the model output.

For instance, to estimate the contrast between `SPHAGNUM` vs `PEAT`, we will inspect the labels of a model that includes substrate type:

```
> coef(m.global.mass)
     (Intercept)     InitMass_cent          InitMass2
    1.1095240564      0.0330830688      -0.0005663352
   SubstrateSOIL SubstrateSPHAGNUM              Shade
    0.1029411303     -0.2053522647      -0.2224253168
```

Thus, we will compute the model-averaged contrast `SPHAGNUM` vs `PEAT` as follows:

```
> modavg(Cand.models, parm = "SubstrateSPHAGNUM")
Multimodel inference on "SubstrateSPHAGNUM" based on AICc

AICc table used to obtain model-averaged estimate:

                  K   AICc Delta_AICc AICcWt Estimate   SE
substrate         4 176.31     172.24      0    -0.28 0.11
shade + substrate 5 172.31     168.24      0    -0.28 0.11
mass + substrate  6  26.82      22.75      0    -0.20 0.06
global            7   4.07       0.00      1    -0.21 0.05
```

```
Model-averaged estimate: -0.21
Unconditional SE: 0.05
95% Unconditional confidence interval: -0.31, -0.1
```

We conclude that mass loss is lower on the *Sphagnum* substrate than on the peat substrate ($\hat{\hat{\beta}}_{SusbstrateSPHAGNUM} = -0.21$, 95% CI: $[-0.31, -0.1]$).

The natural average has been under criticism lately, mainly due to the overestimation of the effect under certain conditions (Cade, 2015). Indeed, excluding models that do not feature the parameter of interest can inflate the model-averaged β, particularly if the parameter only appears in models with low weight. Users should be wary of systematically investigating the effect of parameters appearing in weakly supported models, as using the natural average for this purpose is not recommended. An alternative estimator, the model-averaging estimator with shrinkage, is more robust to this issue (Burnham and Anderson, 2002).

In contrast to the natural average, the model-averaging estimator with shrinkage retains all models in the candidate model set, regardless of the presence of the parameter of interest. Specifically, models without the parameter of interest are assigned a value of 0 for the β and variance. This results in shrinking the effect towards 0 when models without the parameter of interest have high support. Function `modavgShrink( )` implements this approach in `AICcmodavg`:

```
> modavgShrink(cand.set = Cand.models, parm = "Shade")
Multimodel inference on "Shade" based on AICc

AICc table used to obtain model-averaged estimate with shrinkage:

                  K   AICc Delta_AICc AICcWt Estimate   SE
null              2 183.58     179.51      0     0.00 0.00
shade             3 180.21     176.14      0    -0.21 0.09
substrate         4 176.31     172.24      0     0.00 0.00
shade + substrate 5 172.31     168.24      0    -0.22 0.09
mass              4  49.44      45.37      0     0.00 0.00
mass + shade      5  32.32      28.25      0    -0.22 0.05
mass + substrate  6  26.82      22.75      0     0.00 0.00
global            7   4.07       0.00      1    -0.22 0.04

Model-averaged estimate with shrinkage: -0.22
Unconditional SE: 0.04
95% Unconditional confidence interval: -0.31, -0.14


> modavgShrink(Cand.models, parm = "SubstrateSPHAGNUM")
Multimodel inference on "SubstrateSPHAGNUM" based on AICc

AICc table used to obtain model-averaged estimate with shrinkage:

                  K   AICc Delta_AICc AICcWt Estimate   SE
null              2 183.58     179.51      0     0.00 0.00
shade             3 180.21     176.14      0     0.00 0.00
substrate         4 176.31     172.24      0    -0.28 0.11
shade + substrate 5 172.31     168.24      0    -0.28 0.11
```

```
mass               4  49.44      45.37      0     0.00 0.00
mass + shade       5  32.32      28.25      0     0.00 0.00
mass + substrate   6  26.82      22.75      0    -0.20 0.06
global             7   4.07       0.00      1    -0.21 0.05

Model-averaged estimate with shrinkage: -0.21
Unconditional SE: 0.05
95% Unconditional confidence interval: -0.31, -0.1
```

Note that all models are included in the tables above and that the estimate and variance are set to 0 when the parameter does not appear in the model. An additional consideration is that one should strive to balance the number of models with and without the parameter of interest when specifying candidate models. In our case, four models include the effect of shade (vs four without) and four models include the effect of substrate (vs four without).

In our example, both methods of model-averaging β estimates lead to the same conclusions, because the top-ranked model (global model) has all the support and dominates the results. However, whenever several candidate models share the support, both methods of model averaging will not lead to the same conclusions. Model-averaging with shrinkage is the recommended approach for β estimates. A similar approach is also used to model-average predictions.

2.7.2 Inference on predictions

Model-averaging predictions was the originally intended use of model-averaging (Burnham and Anderson, 2002). Function `modavgPred( )` implements this approach, using the entire set of candidate models. As opposed to infering on β estimates, each model can provide an estimate of the prediction, regardless of the presence of a parameter of interest in the model. For instance, both the null and global models can make predictions for the presence of shade, holding the other variables constant. We can illustrate this point using `predict( )`:

```
> ##data frame to make predictions
> ##all variables are held constant, except Shade
> predData <- data.frame(InitMass_cent = c(0, 0),
                         InitMass2 = c(0, 0),
                         Substrate = factor("SOIL",
                                            levels = levels(frog$Substrate)),
                         Shade = c(0, 1))
> ##predictions from global model
> predict(m.global.mass, newdata = predData, se.fit = TRUE)
$fit
        1         2 
1.2124652 0.9900399 

$se.fit
         1          2 
0.05177402 0.04973186 

$df
[1] 115
```

```
$residual.scale
[1] 0.2372509

> ##predictions from null model
> predict(m.null, newdata = predData, se.fit = TRUE)
$fit
        1         2 
0.8235653 0.8235653 

$se.fit
[1] 0.04637383 0.04637383

$df
[1] 120

$residual.scale
[1] 0.5101121
```

The main idea here is that the prediction from the null model does not depend on the frog being in the shade or not. Similarly, we can make predictions for the same two conditions (shade vs no shade) from each model and obtain a model-averaged estimate of the predictions. In other words, the predictions are weighted by the Akaike weights of each model. Consequently, a model with larger weight has a greater influence on the model-averaged prediction than a model with low support. Because `modavgPred( )` relies on `predict( )` methods for different model types, one must supply a `newdata` argument following the same restrictions as for `predict( )`. Specifically, you must supply values for each variable appearing at least once in the candidate models. To assist in this task, the `extractX( )` utility function displays every variable appearing in the model set:

```
> extractX(cand.set = Cand.models)
Predictors appearing in candidate models:
Shade    Substrate    InitMass_cent    InitMass2 

Structure of predictors:
 $ Shade        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Substrate    : Factor w/ 3 levels "PEAT","SOIL",..: 2 3 1 1 2 3 1 2 3 2 ...
 $ InitMass_cent: num  20.36 12.86 5.46 19.26 26.26 ...
 $ InitMass2    : num  414.6 165.4 29.8 371 689.6 ...
```

Using the `predData` data frame above, we proceed with model-averaging predictions:

```
> modavgPred(cand.set = Cand.models, newdata = predData)
Model-averaged predictions on the response scale
based on entire model set and 95% confidence interval:

  mod.avg.pred uncond.se lower.CL upper.CL
1        1.212     0.052    1.111    1.314
2        0.990     0.050    0.893    1.088
```

We can proceed similarly for the substrate type:

```
> ##data frame holding all variables constant, except Substrate
> predSub <- data.frame(InitMass_cent = c(0, 0, 0),
                        InitMass2 = c(0, 0, 0),
                        Substrate = factor(c("PEAT", "SOIL", "SPHAGNUM"),
                                           levels = levels(frog$Substrate)),
                        Shade = c(1, 1, 1))
> ##model-average predictions
> predsMod <- modavgPred(Cand.models, newdata = predSub)
> predsMod
Model-averaged predictions on the response scale
based on entire model set and 95% confidence interval:

  mod.avg.pred uncond.se lower.CL upper.CL
1        0.887     0.047    0.796    0.978
2        0.990     0.050    0.893    1.088
3        0.682     0.049    0.586    0.778
```

To facilitate in the preparation of a plot, I will add the model-averaged predictions, as well as the lower and upper 95% confidence limits in `predSub` to keep everything in the same place. We can first check the content of the `modavgPred` object to reveal the labels for each element we want to extract:

```
> ##check content of object
> str(predsMod)
List of 7
 $ type         : chr "response"
 $ mod.avg.pred : num [1:3] 0.887 0.99 0.682
 $ uncond.se    : num [1:3] 0.0466 0.0497 0.0489
 $ conf.level   : num 0.95
 $ lower.CL     : num [1:3] 0.796 0.893 0.586
 $ upper.CL     : num [1:3] 0.978 1.088 0.778
 $ matrix.output: num [1:3, 1:4] 0.8871 0.99 0.6817 0.0466 0.0497 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:4] "mod.avg.pred" "uncond.se" "lower.CL" "upper.CL"
 - attr(*, "class")= chr [1:2] "modavgPred" "list"
```

Now we can add the elements to the `predSub` data frame:

```
> ##add predictions, lower CL, and upper CL
> predSub$fit <- predsMod$mod.avg.pred
> predSub$low95 <- predsMod$lower.CL
> predSub$upp95 <- predsMod$upper.CL
```

Finally, we can plot the results in a figure:

```
> ##create vector for X axis
> predSub$xvals <- c(0.25, 0.5, 0.75)
> ##create empty box
> plot(fit ~ xvals,
       data = predSub,
       xlim = c(0, 1),
       ylim = range(low95, upp95),
       xlab = "Substrate type",
```

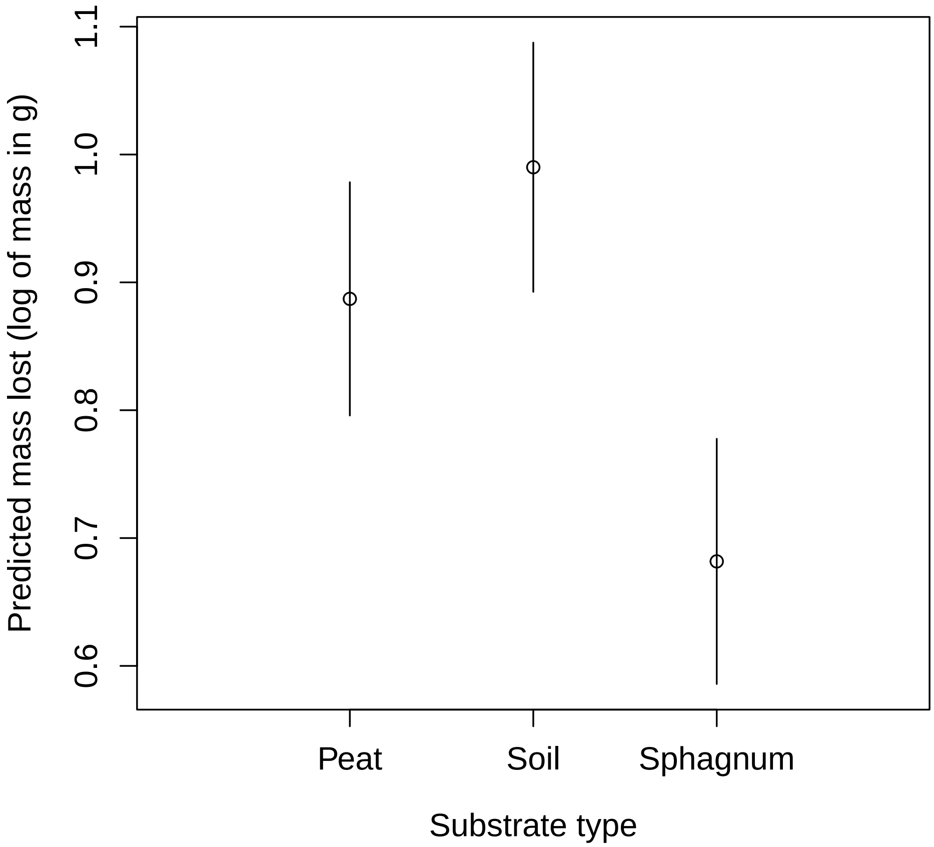

Figure 3: Model-averaged predictions of the log of mass lost by green frogs *Lithobates clamitans* on three different substrate types.

```
        ylab = "Predicted mass lost (log of mass in g)",
        xaxt = "n",
        cex.axis = 1.2,
        cex.lab = 1.2)
> ##add x axis
> axis(side = 1, at = predSub$xvals,
       labels = c("Peat", "Soil", "Sphagnum"),
       cex.axis = 1.2)
> ##add CI's
> segments(x0 = predSub$xvals, x1 = predSub$xvals,
           y0 = predSub$low95, predSub$upp95)
```

The plot clearly shows that mass lost is lower on Sphagnum than on the other substrates (Fig. 3).

Besides model-averaging predictions, it is also possible to model-average effect sizes (differences between groups) using the predictions for two groups from each model. This approach is implemented in the `modavgEffect( )` function. Its use is similar to modavgPred( ), except that the newdata data frame used for prediction must include only two rows (i.e., the two groups to compare). Here is the application on the difference between the peat and Sphagnum substrates:

```
> predComp <- data.frame(InitMass_cent = c(0, 0),
                         InitMass2 = c(0, 0),
                         Substrate = factor(c("PEAT", "SPHAGNUM"),
                                            levels = levels(frog$Substrate)),
                         Shade = c(1, 1))
> ##model-average predictions
> modavgEffect(Cand.models, newdata = predComp)
Model-averaged effect size on the response scale based on entire model set:


Multimodel inference on "SubstratePEAT - SubstrateSPHAGNUM" based on AICc

AICc table used to obtain model-averaged effect size:

                  K   AICc Delta_AICc AICcWt Effect(PEAT - SPHAGNUM)
null              2 183.58     179.51      0                    0.00
shade             3 180.21     176.14      0                    0.00
substrate         4 176.31     172.24      0                    0.28
shade + substrate 5 172.31     168.24      0                    0.28
mass              4  49.44      45.37      0                    0.00
mass + shade      5  32.32      28.25      0                    0.00
mass + substrate  6  26.82      22.75      0                    0.20
global            7   4.07       0.00      1                    0.21
                    SE
null              0.07
shade             0.09
substrate         0.11
shade + substrate 0.12
mass              0.06
mass + shade      0.06
mass + substrate  0.07
global            0.07

Model-averaged effect size: 0.21
Unconditional SE: 0.07
95% Unconditional confidence interval: 0.07, 0.34
```

Again, we conclude that frogs lose mass faster on peat than on Sphagnum substrates. The values reflect differences on the log scale, because we used a log transformation on the original response variable.

3 Support of model classes

3.1 Classes supported

The AICcmodavg package supports models from different model classes. Table 2 illustrates the model types and the model classes compatible with the package.

Table 2: Model classes currently supported by AICcmodavg for model selection and multimodel inference based on AIC, AIC_c, $QAIC$, and $QAIC_c$. Note that support varies from basic model selection to model-averaging predictions.

Model type	Class	Degree of support
beta regression	betareg	model averaging β
conditional logistic regression	clogit	model averaging β
cox proportional hazard	coxph, coxme	model averaging β
distributions	fitdist, fitdistr	model selection
generalized least squares	gls	model averaging β
generalized linear mixed models	glmerMod, glmmTMB	model averaging predictions
latent variable models	lavaan	model selection
linear and generalized linear models	aov, lm, glm, vglm	model averaging predictions
linear mixed models	lme, lmerMod, lmekin, lmerModLmerTest	model averaging predictions
multinomial logistic regression	multinom, vglm	model averaging β
nonlinear model	gnls, nls, nlme, nlmerMod	model selection
occupancy and abundance models with imperfect detectability	unmarkedFit	model averaging predictions
ordinal logistic regression	polr, clm, clmm, vglm	model averaging β
presence-only models	maxlikeFit	model selection
survival regression	survreg	model averaging predictions
zero-inflated models	vglm, zeroinfl	model averaging β
zero-truncated models	vglm, hurdle	model averaging β

The package also offers model selection based on BIC for all model classes in Table 2 with the `bictab( )` function. For Bayesian models of classes `bugs`, `rjags`, or `jagsUI`, model selection using the DIC (Spiegelhalter et al., 2002) is enabled with the `dictab( )` function. A number of functions offer the possibility of conducting model selection and multimodel inference by specifying the basic information (log-likelihood, number of parameters, estimates, standard errors) for models that are not currently supported. The next section features an example using these functions.

3.2 Working with classes not yet supported

Functions `aictabCustom( )` and `bictabCustom( )` perform model selection for model classes not yet supported by `AICcmodavg`. Similarly, `modavgCustom( )` implements multimodel inference for such models. The basic prerequisite is that models be fit with maximum likelihood and that the log-likelihood is available, along with the number of estimated parameters. Of course, the estimates, and their standard errors are required for multimodel inference. In this example, we ran three Cormack-Jolly-Seber capture-mark-recapture models in Program MARK (Cormack, 1964; Jolly, 1965; Seber, 1965;

White and Burnham, 1999). The data were collected during three breeding seasons to investigate the influence of the presence of road-mitigating infrastructures (amphibian tunnels associated with drift fences) on green frog (*Lithobates clamitans*) populations adjacent to roads. We will save the log-likelihoods, number of estimated parameters, and effective sample size in vectors to conduct model selection based on AIC_c:

```
> ##log-likelihoods
> modL <- c(-225.4180, -224.0697, -225.4161)
> ##number of parameters
> modK <- c(2, 3, 3)
> ##model selection
> outTab <- aictabCustom(logL = modL,
                         K = modK,
                         modnames = c("null", "phi(SVL)p(.)",
                                      "phi(Road)p(.)"),
                         nobs = 621)
```

We note that models including the effect of road-mitigating infrastructures have low support compared to models of frog size (SVL: snout-vent length) or the null model. We can also compute the evidence ratio between the top-ranked model vs the model with the effect of road-mitigating infrastructures:

```
> evidence(outTab, model.high = "phi(SVL)p(.)",
           model.low = "phi(Road)p(.)")
```

The top-ranked model has 3.8 times more support than the model with road-mitigating infrastructures. We continue by saving the predicted survival estimates in the presence of road-mitigating infrastructures and their standard errors:

```
> ##survival estimates with road mitigation
> modEst <- c(0.1384450, 0.1266030, 0.1378745)
> ##SE's of survival estimates with road mitigation
> modSE <- c(0.03670327, 0.03347475, 0.03862634)
```

Finally, we compute the model-averaged estimates of survival based on the entire model set:

```
> ##model-averaged survival with road mitigation
> modavgCustom(logL = modL,
               K = modK,
               modnames = c("null", "phi(SVL)p(.)",
                            "phi(Road)p(.)"),
               estimate = modEst,
               se = modSE,
               nobs = 621)
Multimodel inference on manually-supplied parameter based on AICc

 AICc table used to obtain model-averaged estimate:

              K   AICc Delta_AICc AICcWt Estimate   SE
null          2 454.86       0.68   0.36     0.14 0.04
phi(SVL)p(.)  3 454.18       0.00   0.51     0.13 0.03
phi(Road)p(.) 3 456.87       2.69   0.13     0.14 0.04
```

```
Model-averaged estimate: 0.13
Unconditional SE: 0.04
 95 % Unconditional confidence interval: 0.06 , 0.2
```

For comparative purposes, we can compute the model-averaged estimates of survival in the absence of road-mitigating infrastructures:

```
> ##survival estimates without road mitigation
> modEst2 <- c(0.1384450, 0.1266030, 0.1399727)
> ##SE's of survival estimates without road mitigation
> modSE2 <- c(0.03670327, 0.03347475, 0.04981635)
> ##model-averaged survival
> modavgCustom(logL = modL,
               K = modK,
               modnames = c("null", "phi(SVL)p(.)",
                            "phi(Road)p(.)"),
               estimate = modEst2,
               se = modSE2,
               nobs = 621)
Multimodel inference on manually-supplied parameter based on AICc

 AICc table used to obtain model-averaged estimate:

              K   AICc Delta_AICc AICcWt Estimate   SE
null          2 454.86       0.68   0.36     0.14 0.04
phi(SVL)p(.)  3 454.18       0.00   0.51     0.13 0.03
phi(Road)p(.) 3 456.87       2.69   0.13     0.14 0.05

Model-averaged estimate: 0.13
Unconditional SE: 0.04
 95 % Unconditional confidence interval: 0.06 , 0.21
```

Unsuprisingly, we note that survival does not vary with the presence of road-mitigating infrastructures.

The tools in this section highlighted how to conduct model selection and multimodel inference for models that are not yet supported by the package. For models built outside of the R environment in other software and based on maximum likelihood, these tools can be convenient. For model classes that you would wish to be supported by AICcmodavg, you can contact the package author directly.

References

Akaike, H. 1973. Second International Symposium on Information Theory, chapter Information theory as an extension of the maximum likelihood principle, pages 267–281. Akadémiai Kiadó, Budapest, Hungary.

Anderson, A. A. 2019. Assessing statistical results: magnitude, precision, and model uncertainty. American Statistician 73(Suppl.):118–121.

Burnham, K. P. and D. R. Anderson. 1998. Model selection and inference: a practical information-theoretic approach. Springer-Verlag, New York, USA.

Burnham, K. P. and D. R. Anderson. 2002. Model selection and multimodel inference: a practical information-theoretic approach, second edition. Springer-Verlag, New York, USA.

Cade, B. S. 2015. Model averaging and muddled multimodel inferences. Ecology 96:2370–2382.

Calin-Jageman, R. J. and G. Cumming. 2019. The new statistics for better science: ask how much, how uncertain, and what else is known. American Statistician 73(Suppl.):271–280.

Cormack, R. M. 1964. Estimates of survival from the sighting of marked animals. Biometrika 51:429–438.

Dahl, D. B. 2014. xtable: export tables to LaTeX or HTML. R package version 1.7-3. https://cran.r-project.org/package=xtable.

Fiske, I. and R. Chandler. 2011. unmarked: an R package for fitting hierarchical models of wildlife occurrence and abundance. Journal of Statistical Software 43:1–23.

Gelman, A. and J. Hill. 2007. Data analysis using regression and multilevel/hierarchical models. Cambridge University Press, New York, USA.

Goodman, S. N. 1999. Toward evidence-based medical statistics: the P-value fallacy. Annals of Internal Medicine 130:995–1004.

Jolly, G. M. 1965. Explicit estimates from capture-recapture data with both death and immigration: stochastic model. Biometrika 52:225–247.

Lamport, L. 1994. LaTeX: a document preparation system, second edition. Addison-Wesley, New York, USA.

Mazerolle, M. J. 2006. Improving data analysis in herpetology: using Akaike's Information Criterion (AIC) to assess the strength of biological hypotheses. Amphibia-Reptilia 27:169–180.

Mazerolle, M. J. and A. Desrochers. 2005. Landscape resistance to frog movements. Canadian Journal of Zoology 83:455–464.

McCullagh, P. and J. A. Nelder. 1989. Generalized linear models, second edition. Chapman and Hall, New York, USA.

Mittelbach, F. and M. Goossens. 2004. The LaTeX companion, second edition. Addison-Wesley, Boston, MA, USA.

Nuzzo, R. 2014. Statistical errors: P values, the "gold standard" of statistical validity, are not as reliable as many scientists assume. Nature 506:150–152.

Schwarz, G. 1978. Estimating the dimension of a model. Annals of Statistics 6:461–464.

Seber, G. A. F. 1965. A note on the multiple-recapture census. Biometrika 52:249–259.

Spiegelhalter, D. J., N. G. Best, B. P. Carlin, and A. van der Linde. 2002. Bayesian measures of complexity and fit. Journal of the Royal Statistical Society Series B 64:583–639.

Sugiura, N. 1978. Further analysis of the data by Akaike's information criterion and the finite corrections. Communications in Statistics: Theory and Methods A7:13–26.

Wasserstein, R. L., A. L. Schirm, and N. A. Lazar. 2019. Moving to a World beyond "$p < 0.05$". American Statistician 73(Suppl.):1–19.

White, G. C. and K. P. Burnham. 1999. Program MARK: survival estimation from populations of marked animals. Bird Study 46(Suppl.):120–138.