Module 6: Linear models II

Jake Ferguson (jakeferg@hawaii.edu)

Module goals

1. Diagnose and address multicollinearity

1. Select between alternate models

2. Combine information from more than one model

Collinearity versus multicollinearity

Collinearity - when one predictor variable, X_1, is correlated with another, X_2

Multicollinearity - when multiple independent variables are correlated with each other.

Graham 2003. Confronting multicollinearity in ecological multiple regression. Ecology 84:2809-2815.

Examples of collinear predictors

- A person's height and weight
- Monthly average temperature and max/min temperatures
- Mean annual temperature and annual rainfall.

Examples from your study systems?

Type of collinearity

- Variables that are collinear and each have their own separate “effect” on the response variable, Y
  - Tigers avoid areas near human settlements and also areas with domestic animals
  - Domestic animals tend to be found near human settlements
- **Redundant** predictors have essentially the same meaning.
  - various morphometric measures of body size (lengths, masses, ratios, areas)
- **Compositional** variables have to sum to 1 (because last category is determined by others)
  - percent cover of habitat types

Symptoms of predictor collinearity

- Variables may be significant in simple linear regression, but not in multiple regression
- Large standard errors in multiple regression models
- Large changes in coefficient estimates between full and reduced models

Variance Inflation Factors

Get the R^2 coefficient for variable x_j as a function of other predictors:

$$R^2_{x_j} = \text{lm}(x_j \sim x_1 + x_2 + \ldots + x_{j-1} + x_{j+1} + \ldots + x_k)$$

Multicollinearity is then measured using the **variance inflation factor** (VIF):

$$VIF(\hat{\beta}_j) = \frac{1}{1 - R^2_{x_j}}$$

Can be calculated using `vif` in the `car` package.

General rules of thumb:

- Many suggest VIFs > 10 are problematic
- In my experience VIF's > 3 can cause issues

Potential consequences

Models with collinear variables have:

- inflated standard errors
- misleading estimates of effect
- coefficients that are difficult to interpret

Example: Monet

```
##       PRICE HEIGHT WIDTH SIGNED HOUSE   SIZE
## 1 3.993780   21.3  25.6      1     1 545.28
## 2 8.800000   31.9  25.6      1     2 816.64
## 3 0.131694    6.9  15.9      0     3 109.71
## 4 2.037500   25.7  32.0      1     2 822.40
## 5 1.487500   25.7  32.0      1     2 822.40
## 6 1.870000   25.6  31.9      1     1 816.64
```

image: Google Art Project

VIFs

```
library(car)
full.mod ← lm(log(PRICE) ~ HEIGHT + WIDTH + SIZE, data=monet.dat)

vif(full.mod)
```

```
##    HEIGHT     WIDTH      SIZE
##  4.754569  6.013350 13.111192
```

What to do now?

Strategies for Handling Confounding

Consider goals: if **prediction** is your goal collinearity may not be a problem.

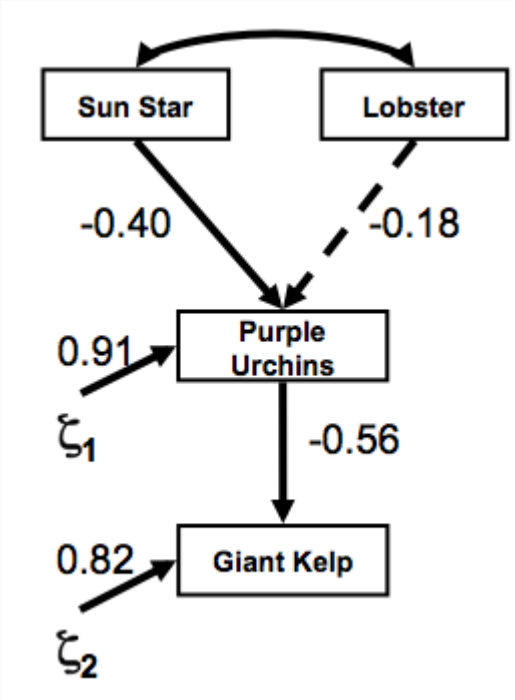
If **understanding** predictor effects is important, consider removing one more variables:

```
nosize.mod ← lm(log(PRICE) ~ HEIGHT + WIDTH, data=monet.dat)
vif(nosize.mod)
```

```
##  HEIGHT   WIDTH
## 1.33921 1.33921
```

Other techniques to handle

- Create new predictors with Principle Components Analysis or Factor analysis
- Ridge regression: gives better estimates of effects and SE's
- Structural equation modeling



Exercise 6A

Examining multicollinearity

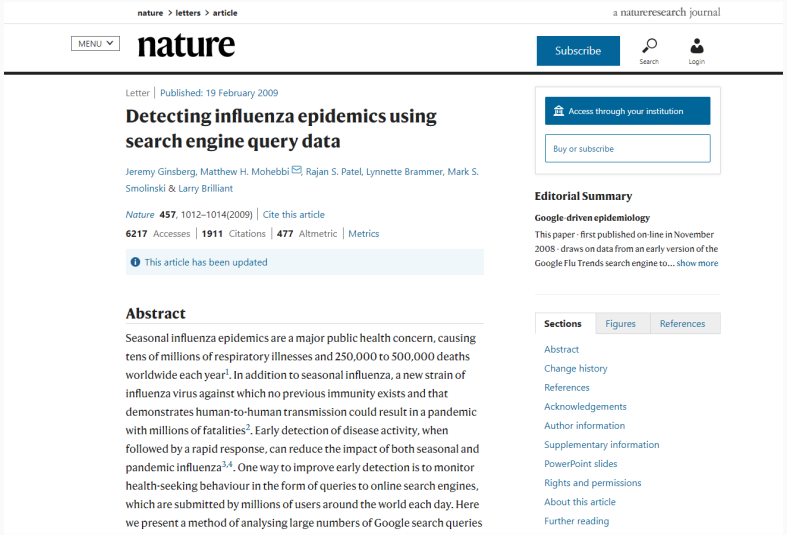
Why do model selection?

- Understand which explanatory variables are important
- Quantify the effects of explanatory variables on the respose
- False belief that it is not legitimate to include ‘insignificant’ regression coefficients.
- May increase predictive precision by dropping unimportant variables

A cautionary example: Google Flu Trends

Why not include all predictors?

- They may be correlated
  - Difficult to determine important effects
- Leads to overfitting, poor prediction



nature > letters > article

a natureresearch journal

MENU nature Subscribe Search Login

Letter | Published: 19 February 2009

Detecting influenza epidemics using search engine query data

Jeremy Ginsberg, Matthew H. Mohebbi, Rajan S. Patel, Lynnette Brammer, Mark S. Smolinski & Larry Brilliant

Nature **457**, 1012–1014(2009) | Cite this article

6217 Accesses | **1911** Citations | **477** Altmetric | Metrics

This article has been updated

Abstract

Seasonal influenza epidemics are a major public health concern, causing tens of millions of respiratory illnesses and 250,000 to 500,000 deaths worldwide each year[1]. In addition to seasonal influenza, a new strain of influenza virus against which no previous immunity exists and that demonstrates human-to-human transmission could result in a pandemic with millions of fatalities[2]. Early detection of disease activity, when followed by a rapid response, can reduce the impact of both seasonal and pandemic influenza[3,4]. One way to improve early detection is to monitor health-seeking behaviour in the form of queries to online search engines, which are submitted by millions of users around the world each day. Here we present a method of analysing large numbers of Google search queries

Access through your institution

Buy or subscribe

Editorial Summary

Google-driven epidemiology

This paper · first published on-line in November 2008 · draws on data from an early version of the Google Flu Trends search engine to... show more

Sections | Figures | References

Abstract
Change history
References
Acknowledgements
Author information
Supplementary information
PowerPoint slides
Rights and permissions
About this article
Further reading

https://www.wired.com/2015/10/can-learn-epic-failure-google-flu-trends/

How to compare models?

- Look at correlations between individual predictors and the response variable
- Look at the p-value from the ANOVA F-test of overall model fit
- Compare R^2 or R^2_{adj} between competing models
- Assess ability to predict a new dataset

Note that these different criterion will often point to different models as best!

Example: Sleep in mammals

- Lifespan (*years*)
- Gestation (*days*)
- log(Brain weight) (*g*)
- log(Body Weight) (*kg*)
- Predation Index (1-5; 1 = least likely to be preyed upon)
- Exposure Index [1-5: 1 = least exposed (e.g., animal sleeps in a den)]
- Danger Index (1:5, combines exposure and predation; 1= least danger from other animals)

Any guesses which are most highly associated with total sleep?

Allison, Truett and Cicchetti, Domenic V. (1976), Sleep in Mammals: Ecological and Constitutional Correlates, Science, November 12, vol. 194

Model 1: Sleep and Lifespan

OK, `LifeSpan` is significant

```
summary(lm(TotalSleep ~ LifeSpan, data=sleep.dat))
```

```
##
## Call:
## lm(formula = TotalSleep ~ LifeSpan, data = sleep.dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.2781 -2.7714  0.3248  2.4586  9.5964
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.27314    1.01443  12.099 7.17e-14 ***
## LifeSpan    -0.08206    0.04117  -1.993   0.0543 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.46 on 34 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.1046,    Adjusted R-squared:  0.07831
## F-statistic: 3.974 on 1 and 34 DF,  p-value: 0.05429
```

Model 2: Sleep and LifeSpan + Danger

OK, `LifeSpan` is still significant

```
summary(lm(TotalSleep ~ LifeSpan + Danger, data=sleep.dat))
```

```
##
## Call:
## lm(formula = TotalSleep ~ LifeSpan + Danger, data = sleep.dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4670 -1.5811  0.0289  2.1701  6.2612
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 18.03550    1.30037  13.870 2.54e-15 ***
## LifeSpan    -0.09329    0.03046  -3.063  0.00435 **
## Danger      -2.15770    0.39806  -5.421 5.33e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.293 on 33 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.5264,    Adjusted R-squared:  0.4977
```

Model 3: Sleep and LifeSpan + Danger +

Snap!

```
summary(lm(TotalSleep ~ LifeSpan + Danger + log(BrainWt), data=sleep.dat))
```

```
##
## Call:
## lm(formula = TotalSleep ~ LifeSpan + Danger + log(BrainWt), data = sleep.dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8545 -1.2643 -0.2213  2.1568  4.8204
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  18.68659    1.20426  15.517  < 2e-16 ***
## LifeSpan     -0.01554    0.03908  -0.398   0.6936
## Danger       -1.99159    0.36656  -5.433 5.63e-06 ***
## log(BrainWt) -0.85280    0.30252  -2.819   0.0082 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Is LifeSpan an important predictor?

T-test

Model 1:

```
##                Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 12.27314106 1.01443011 12.098558 7.173668e-14
## LifeSpan    -0.08206275 0.04116776 -1.993374 5.429371e-02
```

Model 3:

```
##                 Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)  18.68658631 1.20425929 15.5170788 1.908941e-16
## LifeSpan     -0.01553554 0.03908285 -0.3975027 6.936360e-01
## Danger       -1.99158708 0.36656242 -5.4331458 5.632896e-06
## log(BrainWt) -0.85279547 0.30251767 -2.8189939 8.196780e-03
```

Correlated predictors can strongly influence estimates and significance!

Hypothesis test of nested models

```
mod2 ← lm(TotalSleep ~ Danger + log(BrainWt), data=sleep.dat, na.action=na.exclude)
mod3 ← lm(TotalSleep ~ log(BrainWt) + Danger + log(BodyWt), data=sleep.dat, na.actior

anova(mod2, mod3)
```

```
## Analysis of Variance Table
##
## Model 1: TotalSleep ~ Danger + log(BrainWt)
## Model 2: TotalSleep ~ log(BrainWt) + Danger + log(BodyWt)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     35 311.65
## 2     34 304.55  1    7.0961 0.7922 0.3797
```

Model 2 and 3 are not significantly different.

But this won't let us compare more than two models!

Adjusted R^2

$$R^2 = 1 - \frac{\text{SSR}}{\text{SST}}$$

$$R^2_{adj} = 1 - \frac{\text{SSR}/(n-k-1)}{\text{SST}/(n-1)}$$

n = sample size

k = number of parameters

- R^2 always increases with the number of predictors
- R^2_{adj} takes into account the number of predictors used
- As the number of explanatory variables increases, R^2_{adj} is less than R^2
- High values of R^2_{adj} can be used to choose the best model
    - Can be done with a large set of models

Adjusted R^2 example

```
summary(mod2)$adj.r.squared
```

```
## [1] 0.5640078
```

```
summary(mod3)$adj.r.squared
```

```
## [1] 0.5614038
```

Suggests that model 2 better explains the data.

Potential issues: R^2_{adj} has high variability, so results are dependent on the data. Probability of selecting a suboptimal model is often high.

Exercise 6B

F-tests and adjusted R^2

Trade-offs in model fit and complexity

Including lots of predictors increases the amount of variation you explain in your dataset (aka the training data in blue), but does a poor job of explaining variation in a new dataset (aka the test data in red).

Ideally the R^2 value in the training and the test data are the same.

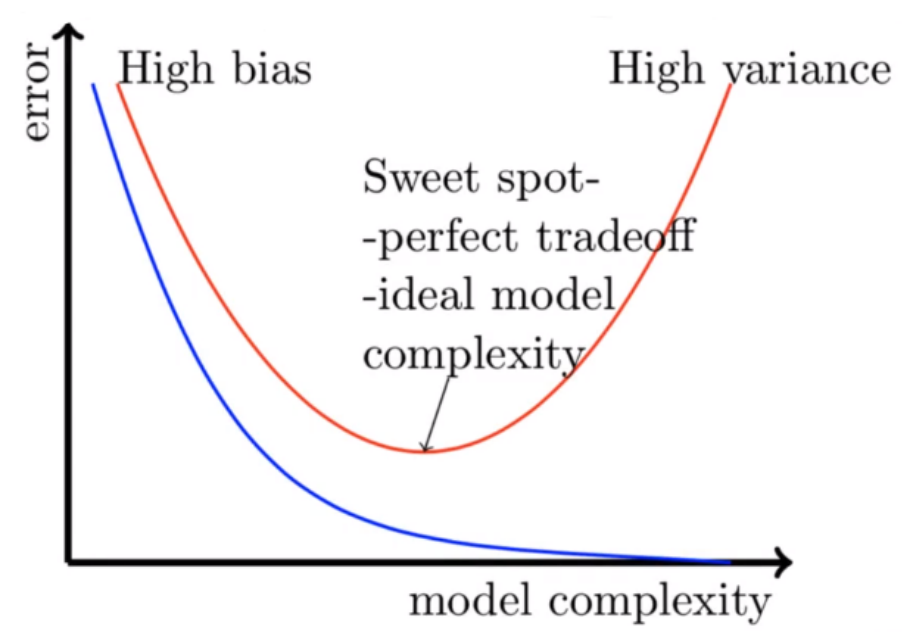


image: medium.com

Assessing prediction

A machine learning approach

1. Break up data into **training** and **test** datasets
    - typically 2/3 and 1/3 of dataset
2. Fit the model to the training data
3. Use the fits to predict the test data, calculate RMSE

Example

```
train.dat ← sleep.dat[1:40,]
test.dat ← sleep.dat[41:62,]

mod2.train ← lm(TotalSleep ~ Danger + log(BodyWt), data=train.dat)
mod3.train ← lm(TotalSleep ~ log(BrainWt) + Danger + log(BodyWt), data=train.dat)
```

```
sqrt(mean((test.dat$TotalSleep - predict(mod2.train, newdata=test.dat))^2, na.rm=TRUE)
```

```
## [1] 6.348124
```

```
sqrt(mean((test.dat$TotalSleep - predict(mod3.train, newdata=test.dat))^2, na.rm=TRUE)
```

```
## [1] 4.271833
```

Here model 3 is better!

But for small datasets like this, our conclusion might change with a different partition of the test and training data.

Leave-one-out Cross-Validation

This is the recipe for leave-one-out cross validation:

1. Fit all the data except for one datapoint
2. Predict the held-out datapoint, $\hat{Y}$
3. Calculate the error between the predicted and held-out value, $\hat{Y} - Y$.
4. Go back and do it for the next datapoint

Implementing in R

```
library(caret)

sleep.control ← trainControl(method = "LOOCV") #tell the function we want to do leav

mod2.xv ← train(TotalSleep ~ Danger + log(BodyWt), data=sleep.dat, method = "lm", na
mod3.xv ← train(TotalSleep ~ log(BrainWt) + Danger + log(BodyWt), data=sleep.dat, me

print(mod2.xv$results)
```

```
##   intercept     RMSE  Rsquared      MAE
## 1      TRUE 3.258317 0.4688809 2.666278
```

```
print(mod3.xv$results)
```

```
##   intercept     RMSE  Rsquared      MAE
## 1      TRUE 3.128541 0.5108055 2.555539
```

Exercise 6C

LOOCV

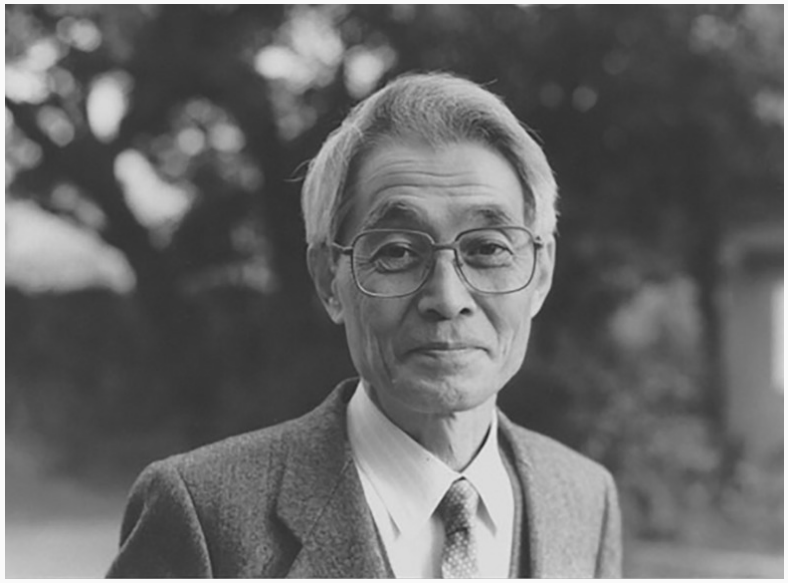
Here we predict model 3 is better again.

However, what is the uncertainty in the RMSE?

Akaike's Information Criterion

If models are not nested or you need to compare more than two models pairwise comparisons won't work

Hirotugu Akaike estimated the bias in the estimate of the likelihood, finding that bias increased with the number of model parameters.



Cavanauh, J.E. and Neath, A.A. (2019) The Akaike information criterion: Background, derivation, properties, application, interpretation, and refinements. Wiley Interdisciplinary Reviews.

The AIC

AIC = $-2\ell(\hat{\theta}; Y) + 2k$

AICc = $-2\ell(\hat{\theta}; Y) + 2k + \frac{2k^2+2k}{n-k-1}$

These criterion tradeoff the model **fit** (the likelihood of the model $\ell(\hat{\theta}; Y)$) and the model **complexity** (the number of parameters, k).

- Fit all models, the lowest AIC (AICc) value is the best model.
- Larger differences in AIC (AICc), called ΔAIC, are stronger support for the best model.
- Rules of thumb for support of suboptimal model
  - 0- 7: plausible
  - 7-14: equivocal
  - $>$ 14: impossible

Burnham, Anderson, Huyvaert (2011) Behavioral Ecology and Sociobiology

Example

```
library(MuMIn)
model.sel(list(mod2, mod3), rank=AIC)
```

```
## Model selection table
##   (Int)    Dng log(BrW) log(BdW) df  logLik   AIC delta weight
## 1 18.12 -1.861  -0.9035           4 -93.901 195.8  0.00  0.637
## 2 19.35 -1.885  -1.4650   0.4807  5 -93.464 196.9  1.12  0.363
## Models ranked by AIC(x)
```

AICc selects model 2 but ΔAIC = 1.03.

Fitting more models

```
mod1 ← lm(TotalSleep ~ log(BrainWt), data=sleep.dat)
mod2 ← lm(TotalSleep ~ Danger + log(BrainWt), data=sleep.dat)
mod3 ← lm(TotalSleep ~ LifeSpan, data=sleep.dat)
mod4 ← lm(TotalSleep ~ log(BrainWt) + Danger + log(BodyWt) + LifeSpan, data=sleep.dat
mod5 ← lm(TotalSleep ~ log(BrainWt) + Danger + log(BodyWt) + LifeSpan + Exposure, dat
mod6 ← lm(TotalSleep ~ log(BrainWt) + Danger + log(BodyWt) + LifeSpan*Exposure, data=

model.sel(list(mod1, mod2, mod3, mod4, mod5, mod6), rank=AIC)
```

```
## Model selection table
##   (Int) log(BrW)    Dng       LfS log(BdW)    Exp   Exp:LfS df   logLik   AIC
## 4 19.41  -1.2450 -1.990 -0.008893   0.3063                   6  -88.266 188.5
## 5 19.56  -1.3570 -2.394 -0.008660   0.2541 0.5591             7  -87.909 189.8
## 6 19.57  -1.3520 -2.393 -0.009335   0.2504 0.5479 0.0004399   8  -87.909 191.8
## 2 18.12  -0.9035 -1.861                                      4  -93.901 195.8
## 3 12.27                 -0.082060                            3 -103.881 213.8
## 1 13.63  -0.9983                                             3 -104.941 215.9
##   delta weight
## 4  0.00  0.573
## 5  1.29  0.301
## 6  3.29  0.111
## 2  7.27  0.015
## 3 25.23  0.000
```

Exercise 6D

Model selection with AIC

Model averaging

Rather than choose a single best model, another approach is to average predictions among "competitive" models or models with roughly equal support.

$$\hat{\theta}_{avg} = \sum W_i \theta_i$$

Calculate a standard error that accounts for model uncertainty and sampling uncertainty:

$$\widehat{SE}_{avg} = \sum_i W_i \sqrt{SE_i^2 + (\hat{\theta}_{avg} - \hat{\theta}_i)^2}$$

Typically, 95% CIs are formed using $\hat{\theta}_{avg} \pm 1.96 \cdot \widehat{SE}_{avg}$.

AIC weights

1. Compute model weights, using the AIC values, reflecting relative plausibility of the different models

$$W_i = \frac{exp^{-\Delta \mathrm{AIC_i}}}{\sum_j exp^{-\Delta \mathrm{AIC_j}}}$$

2. Calculate weighted predictions and SEs that reflect model uncertainty and sampling uncertainty

Example

```
AIC.list ← model.sel(list(mod1, mod2, mod3, mod4, mod5, mod6), rank=AIC)

coef(model.avg(AIC.list)) #extract model averaged estimates
```

```
##       (Intercept)      log(BrainWt)            Danger       log(BodyWt)
##     19.4531688312     -1.2852080307     -2.1544877453      0.2840339391
##          LifeSpan          Exposure Exposure:LifeSpan
##     -0.0088718739      0.5560591092      0.0004399214
```

```
sqrt(diag(vcov(model.avg(AIC.list)))) #model averaged standard errors
```

```
##       (Intercept)      log(BrainWt)            Danger       log(BodyWt)
##        1.85015203        0.81763865        0.54150786        0.59308295
##          LifeSpan          Exposure Exposure:LifeSpan
##        0.04447710        0.82645576        0.03011468
```

```
#can also print summary(model.avg( ... )) but too long for here.
```

1. Define a set of *a-priori* models
2. Fit & calculate AIC values/weights

Resources

- Ken Burham & David Anderson, Model Selection and Multimodel Inference. A classic book
- Mark Brewer (Model selection and the cult of AIC) [https://www.youtube.com/watch?v=lEDpZmq5rBw]
- R package `caret` for cross-validation (**C**lassification **A**nd **RE**gression **T**raining)
- R package `MuMIn` for model averaging
- Some issues with using model averaging for explanatory modeling are discussed in Cade, B. S. (2015). Model averaging and muddled multimodel inferences. Ecology, 96(9), 2370-2382.
- Strengths and weaknesses of R^2_{adj}: https://davegiles.blogspot.com/2013/08/unbiased-model-selection-using-adjusted.html