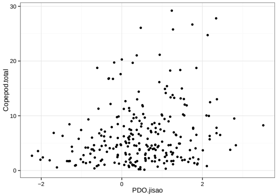
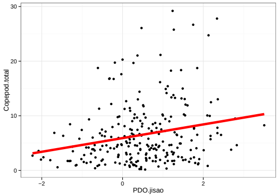
Information Theoretic Approaches to Model Selection

What is the Shape of this Relationship?



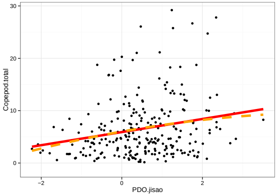
What is the Shape of this Relationship?

Is it linear?



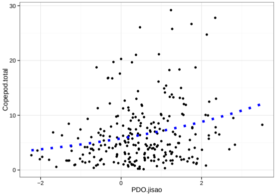
What is the Shape of this Relationship?

Is is squared?

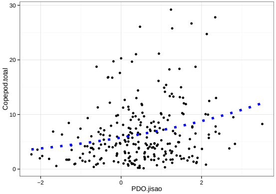


What is the Shape of the Relationship?

Is it exponential with a Gamma error?



What is the *R*elative Support for Each Relationship?



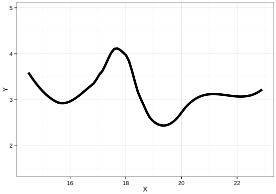
Model Selection in a Nutshell

The Frequentist P-Value testing framework emphasizes the evaluation of a single hypothesis - the null. We evaluate whether we reject the null.

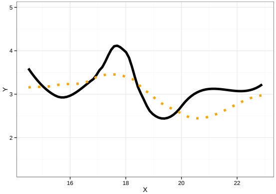
This is perfect for an experiment where we are evaluating clean causal links, or testing for a a predicted relationship in data.

Often, though, we have multiple non-nested hypotheses, and wish to evaluate each. To do so we need a framework to compare the relative amount of information contained in each model and select the best model or models. We can then evaluate the individual parameters.

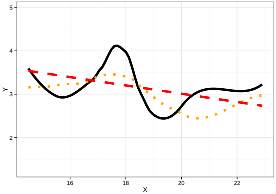
Suppose this is the Truth



We Can Fit a Model To Descibe Our Data, but it Has Less Information



We Can Fit a Model To Descibe Our Data, but it Has Less Information



Information Loss and Kullback-Leibler Divergence

$$I(f,g) = \int f(x) log \frac{f(x)}{g(x|\theta)} dx$$

where I(f,g) = information loss when a function g is used to approximate the truth, f - integrated over all values of x when g is evaluated with some set of parameters θ

Two neat properties:
1) We can re-arrange to pull out a term $-log(g(x|\theta))$ which is our negative Log-Likelihood!
2) If we want to compare the relative loss of $I(f,g_1)$ and $I(f,g_2)$, f(x) drops out as a constant!

Defining an Information Criterion

$$I(f,g) + constant = -log(L(\theta|x)) + K$$

where K is the number of parameters for a model

This gives rise to Akaike's Information Criterion - lower AIC means less information is lost by a model

$$\boldsymbol{AIC = -2log(L(\theta|x)) + 2K}$$

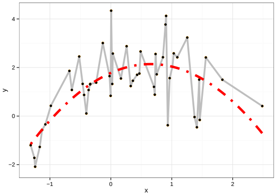
Balancing Fit and Parsimony

A model with n-1 parameters from a dataset with n points with always fit your data perfectly, but -

- With more parameters, variance in the estimates of parameters can become inflated
- But, with too few parameters, estimates of parameters become biased.

Balancing General and Specific Truths

Which model better describes a general principle of how the world works?



How many parameters does it take to draw an elephant?

But Sample Size Can Influence Fit...

$$AIC = -2log(L(\theta|x)) + 2K$$

$$AICc = AIC + \frac{2K(K+1)}{n-K-1}K$$

Variations on a Theme: Other IC Measures

For overdispersed count data, we need to accomodate the overdispersion parameter

$$QAIC = \frac{-2log(L(\theta|x))}{\hat{c}} + 2K$$

where $\hat{c}$ is the overdispersion parameter

Many other IC metrics for particular cases that deal with model complexity in different ways. For example

$$BIC = -2log(L(\theta|x)) + Kln(n)$$

Implementing AIC: Create Models

```
cop_linear <- glm(Copepod.total ~ PDO.jisao , data=plankton)
#
cop_square <- glm(Copepod.total ~ poly(PDO.jisao,2), data=plankton)
#
cop_glm <- glm(Copepod.total ~ PDO.jisao , data=plankton,
               family=Gamma(link="log"))
```

Implementing AIC: Compare Models

```
AIC(cop_linear)

# [1] 1617

AIC(cop_square)

# [1] 1618
```

```
AIC(cop_square)

# [1] 1618

AIC(cop_glm)

# [1] 1459
```

How can we Use AIC Values?

$$\Delta AIC = AIC_i - min(AIC)$$

Rules of Thumb from Burnham and Anderson(2002):
Δ AIC ¡ 2 implies that two models are similar in their fit to the data
Δ AIC between 3 and 7 indicate moderate, but less, support for retaining a model
Δ AIC ¿ 10 indicates that the model is very unlikely

A Quantitative Measure of Relative Support

$$w_i = \frac{e^{\Delta_i/2}}{\sum_{r=1}^{R} e^{\Delta_i/2}}$$

Where w_i is the *r*elative support for model i compared to other models in the set being considered.

Model weights summed together $= 1$

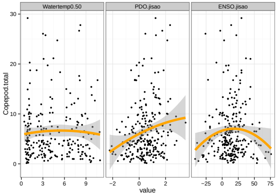
Model Weight Comparison

```
copepodList <- list(cop_linear, cop_square, cop_glm)
names(copepodList) <- c("linear", "square", "Gamma-log")
#
aictab(cand.set = copepodList,
       modnames = names(copepodList))
```

Model Weight Comparison

```
#
# Model selection based on AICc :
#
#           K AICc Delta_AICc AICcWt Cum.Wt     LL
# Gamma-log 3 1459        0.0      1      1 -726.6
# linear    3 1617      157.5      0      1 -805.4
# square    4 1619      159.3      0      1 -805.2
```

What if You Have a LOT of Hypotheses?



7 models alone if we keep linear and squared terms grouped

Exercise: Construct an AICc Table

```
full_lm0 <- lm(Copepod.total ~ Watertemp0.50 +
                 I(Watertemp0.50^2) +
                PDO.jisao + I(PDO.jisao^2) +
                ENSO.jisao+ I( ENSO.jisao^2),
              data=plankton)
```

Use this model as a jumping off point, and construct a series of nested models with subsets of the variables. Evaluate using AICc Weights

All 8 (intercept only!) Models

```
aictab(modList, modnames=names(modList))

#
# Model selection based on AICc :
#
#                  K AICc Delta_AICc AICcWt Cum.Wt     LL
# Full Model       8 1463       0.00   0.72   0.72 -723.0
# No ENSO          6 1465       2.01   0.26   0.98 -726.1
# No PDO           6 1472       8.92   0.01   0.99 -729.6
# Temperature Only 4 1472       9.14   0.01   1.00 -731.8
# No Temperature   6 1616     153.25   0.00   1.00 -801.8
# PDO Only         4 1619     156.06   0.00   1.00 -805.2
# ENSO Only        4 1627     163.96   0.00   1.00 -809.2
# Null             2 1628     165.50   0.00   1.00 -812.0
```

Variable Weights

Variable Weight = sum of all weights of all models including a variable. Relative support for inclusion of parameter in models.

```
importance(modList, parm="ENSO.jisao", modnames=names(modList))

#
# Importance values of ' ENSO.jisao ' :
#
# w+ (models including parameter):  0.73
# w- (models excluding parameter):  0.27
```

Model Averaged Parameters

$$\hat{\hat{\beta}} = \frac{\sum w_i \hat{\beta}_i}{\sum w_i}$$

$$var(\hat{\hat{\beta}}) = \left[w_i \sqrt{var(\hat{\beta}_i) + (\hat{\beta}_i - \hat{\hat{\beta}}_i)^2} \right]^2$$

Buckland et al. 1997

Model Averaged Parameters

```
#
# Multimodel inference on " ENSO.jisao " based on AICc
#
#  AICc table used to obtain model-averaged estimate:
#
#                K AICc Delta_AICc AICcWt Estimate   SE
# Full Model     8 1463       0.00   0.99    -0.03 0.02
# No PDO         6 1472       8.92   0.01     0.01 0.02
# No Temperature 6 1616     153.25   0.00    -0.03 0.02
# ENSO Only      4 1627     163.96   0.00     0.01 0.02
#
# Model-averaged estimate: -0.03
# Unconditional SE: 0.02
#  95 % Unconditional confidence interval: -0.08 , 0.01
```

Model Averaged Predictions

```
newData <- data.frame(Watertemp0.50 = 3,
                      PDO.jisao=0.2,
                      ENSO.jisao=25)
#
modavgpred(modList, modnames=names(modList), newdata = newData)

#
# Model-averaged predictions on the response scale based on entire mode
#
#   mod.avg.pred uncond.se
# 1         6.17      0.69
```

95% Model Confidence Set

```
confset(modList, modnames=names(modList))

#
# Confidence set for the best model
#
# Method:  raw sum of model probabilities
#
# 95% confidence set:
#            K AICc Delta_AICc AICcWt
# Full Model 8 1463       0.00   0.72
# No ENSO    6 1465       2.01   0.26
#
# Model probabilities sum to 0.98
```

Renormalize weights to 1 before using confidence set for above model averaging techniques

Cautionary Notes

- AIC analyses aid in model selection. One must still evaluate parameters and parameter error.
- Your inferences are constrained solely to the range of models you consider. You may have missed the 'best' model.
- All inferences **MUST** be based on *a* priori models. Post-hoc model dredging could result in an erroneous 'best' model suited to your unique data set.

But...

Considering MANY subsets of a larger model is tedious. Computational methods can speed the way. Calcagno's `glmulti` package provides a flexible framework for multi-model consideration.

```
library(glmulti)
full_glmulti <- glmulti(full_lm, level=1, plot=F)

# Initialization...
# TASK: Exhaustive screening of candidate set.
# Fitting...
#
# After 50 models:
# Best model: Copepod.total~1+I(cent(Watertemp0.50)^2)+PDO.jisao+I(cent
# Crit= 1459.02599976963
# Mean crit= 1507.78237770236
# Completed.
```

Model Averaged Coefficients

```
coef(full_glmulti)

#                              Estimate Uncond. variance
# I(cent(PDO.jisao)^2)        0.0010693        5.277e-03
# ENSO.jisao                 -0.0193893        5.268e-04
# I(cent(Watertemp0.50)^2)   -0.0120659        9.966e-04
# Watertemp0.50               0.0434154        9.129e-03
# I(cent(ENSO.jisao)^2)      -0.0007269        4.347e-07
# PDO.jisao                   1.3770654        1.996e-01
# (Intercept)                 6.2175040        4.915e-01
....
```

Multiple SE methods implemented

Model Averaged Coefficients

```
coef(full_glmulti)

....

#                            Nb models  Importance
# I(cent(PDO.jisao)^2)              32      0.2742
# ENSO.jisao                        32      0.5755
# I(cent(Watertemp0.50)^2)          32      0.5842
# Watertemp0.50                     32      0.6077
# I(cent(ENSO.jisao)^2)             32      0.6984
# PDO.jisao                         32      0.9921
# (Intercept)                       64      1.0000
....
```

Multiple SE methods implemented

Model Averaged Coefficients

```
coef(full_glmulti)

....

#                              +/- (alpha=0.05)
# I(cent(PDO.jisao)^2)                0.143142
# ENSO.jisao                          0.045225
# I(cent(Watertemp0.50)^2)            0.062205
# Watertemp0.50                       0.188265
# I(cent(ENSO.jisao)^2)               0.001299
# PDO.jisao                           0.880284
# (Intercept)                         1.381330
```

Multiple SE methods implemented

Comparison of Predictions from Full Model to Full MMI

```
as.data.frame(predict(full_lm, newdata=newData, se.fit=T))

#     fit se.fit  df residual.scale
# 1 6.083 0.7009 226           5.47

modavgpred(modList, modnames=names(modList), newdata = newData)

#
# Model-averaged predictions on the response scale based on entire model set:
#
#   mod.avg.pred uncond.se
# 1         6.17      0.69

as.data.frame(predict(full_glmulti, newdata=newData, se.fit=T))

#      X1 variability.Uncond..variance
# 1 6.138                       0.4606
#   variability......alpha.0.05. omittedNA
# 1                        1.337         0
```

Exercise: Consider the Diatom

```
diatom_lm <- lm(diatom ~ Copepod.total * Bosmina...Daphnia *
                    Watertemp0.50, data=plankton)
```

- Examine the data and consider valid model choices
- Fit models, and evalute variable importance
- What model(s) have good support?
- What parameter(s) have good support?