Statistical models: linking data to theory

Model comparison and selecion

Sara Mortara

3 Fev 2025

Recap: How do we link models and data?

- We use statistical models to formally represent an hypothesis
- General recipe for building models:

1. Identifying variables – Distinguishing observable variables (data) from unobservable ones (parameters)
2. Defining relationships – Expressing each variable in terms of others or probability distributions
3. Building the generative model – Combining variables and distributions to simulate and analyze data

Recap: How do we link models and data?

- We follow a language for describing statistical models

$$y_i \sim Normal(\mu, sigma)$$

$$\mu_i = \beta x_i$$

$$\beta \sim Normal(0, 10)$$

$$\sigma \sim Exponential(1)$$

Why a generative model?

A Bayesian model is called a generative model because it defines a probabilistic process that can generate data. This means it specifies the joint probability distribution of both observed data and unobserved parameters

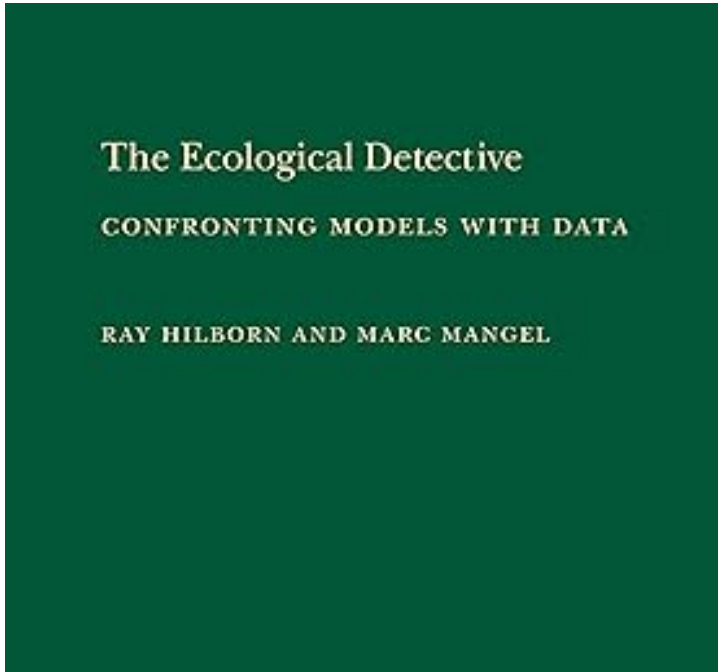
This allows us to:

Simulate Data – Given parameter values, we can generate hypothetical datasets that resemble real-world observations

Infer Parameters – Given observed data, we update our beliefs about the parameters using Bayes' theorem

Answering questions in Ecology: Ecological detective

How do we confront multiple hypotheses with data and assign degrees of belief to different hypotheses?



Beyond *How do we link models and data?*

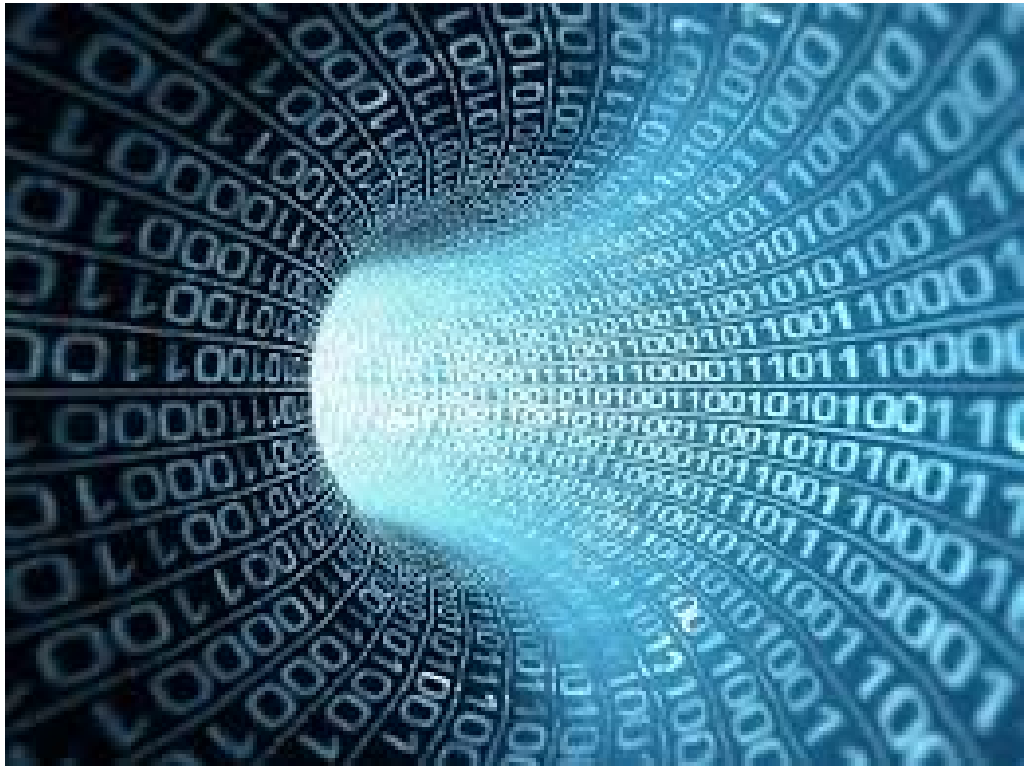
The tools of the ecological detective

- Hypotheses
- Data
- Goodness of fit
- Numerical procedures

Hypotheses

- Science consists of confronting different descriptions of how the world works with data
- We use data to arbitrate between different descriptions
- We use the "best" description to make additional predictions and decisions
- If we translate different hypotheses into quantitative predictions via models we can simultaneously confront all of them

Data



- Represents a particular view of the world
- Know your data
- Define the likelihood of your data

Data

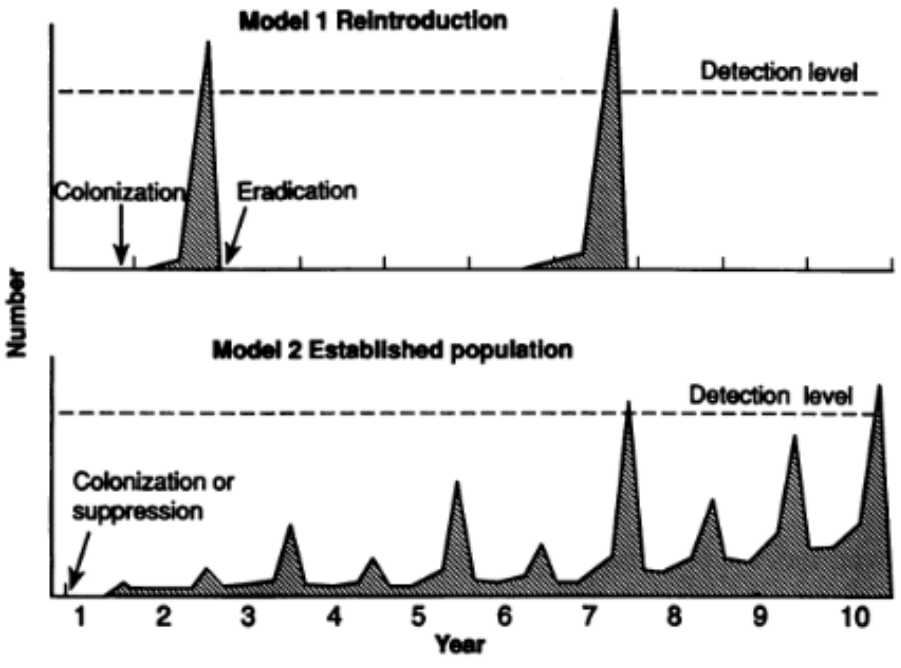
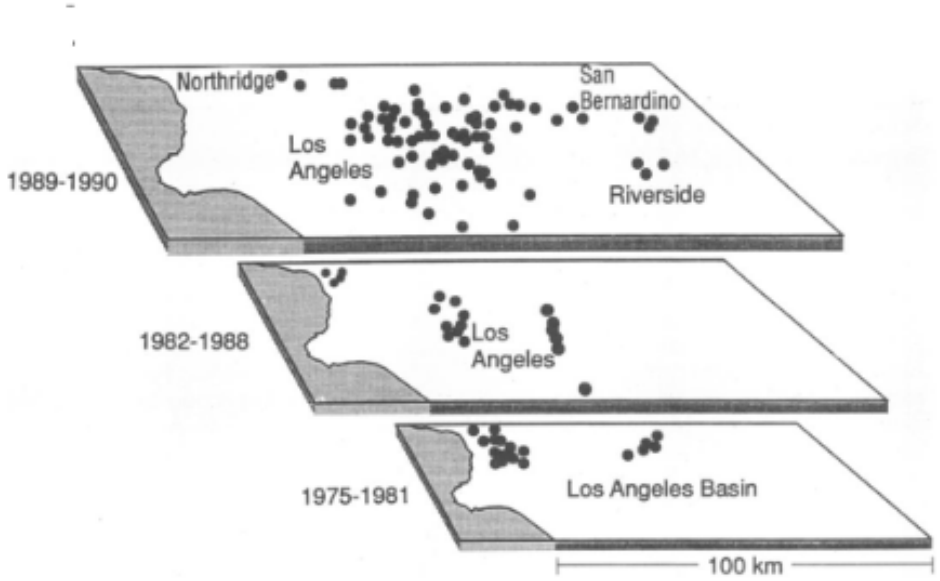


Fig. 1. Diagram illustrating patterns of medfly colonization and growth according to two alternative invasion models.

Carey 1991

- Mediterranean fruit fly (medfly) *Creatitis capitata*
- Destructive agricultural pest
- Climatic and host contitions are right for its establishment in California
- Sporadic outbreaks over decades
- New colonization event or medfly is established below the level for detection?

Data



Carey 1991

- Medfly captures from 1975-1990
- Each point represents a location of medfly captures
- Intervals between captures are decreasing
- Area over which they are detected is expanding
- Evidence that previous eradication programs did not eradicate the medfly from California

Goodness of fit

- Data are used to arbitrate between different hypotheses or models
- Measure of how each description of the world fits the observations

  MSE: Squared error between predictions & observed data

Numerical procedures

Emphasize predictive performance rather than just model fit to observed data

Posterior Predictive Checks: Simulated Data Replication: Generate synthetic datasets using posterior samples and compare them to observed data

Cross-validation

- LOO-CV (Leave-One-Out Cross-Validation): Uses Pareto-smoothed importance sampling (PSIS-LOO) to estimate predictive accuracy by leaving out one observation at a time

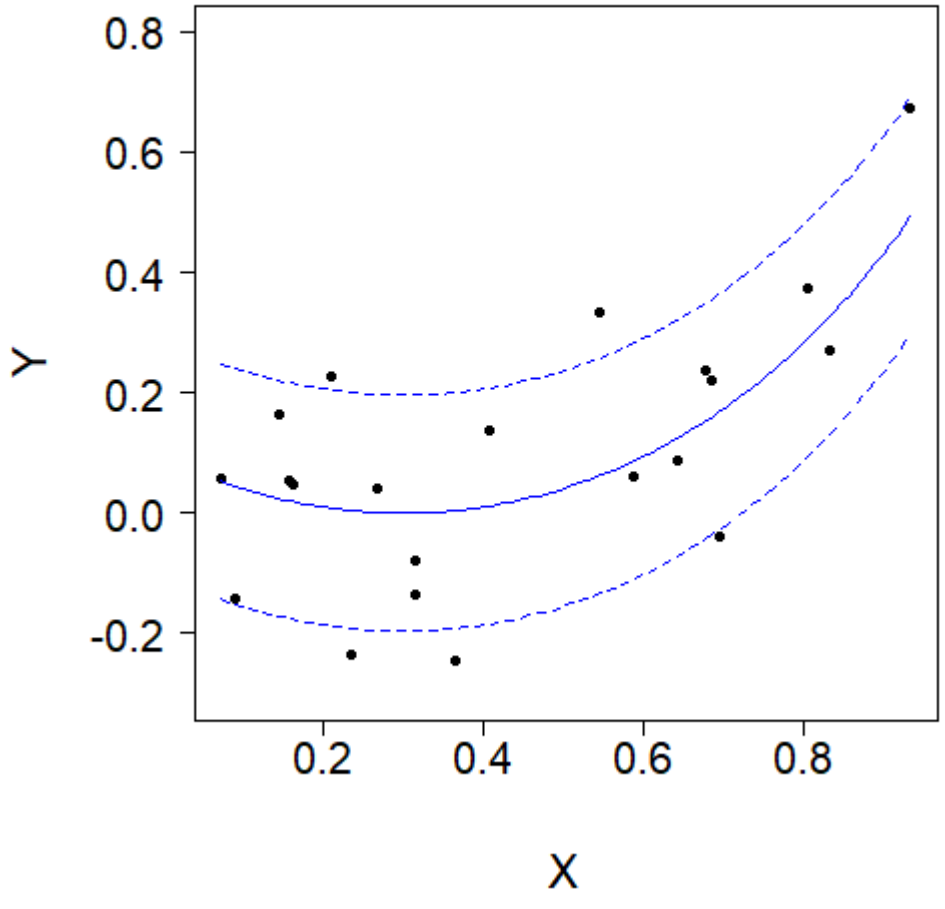
Information criteria

- Akaike Information Criterion (AIC): Balances model fit with complexity by penalizing the number of parameters
- WAIC (Widely Applicable Information Criterion): A Bayesian alternative to AIC that estimates out-of-sample predictive accuracy using the log pointwise posterior predictive density (lppd)

Inferences about competing models

The data-generating process (a.k.a. true model):

$$y_i \sim \mathcal{N}(\mu_i, \sigma)$$
$$\mu_i = e^{(x_i - 0.3)^2} - 1$$
$$\sigma = C$$

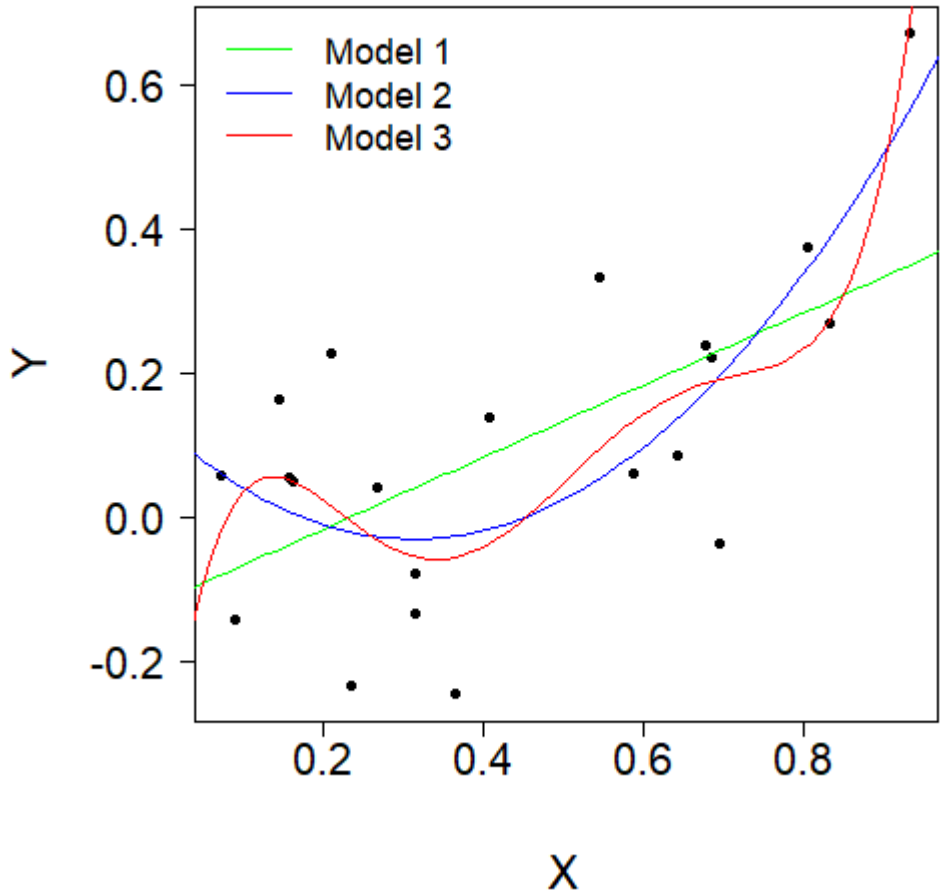


The models

$$\mathbf{M1}: \mu_i = \alpha + \beta_1 x_i$$

$$\mathbf{M2}: \mu_i = \alpha + \beta_1 x_i + \beta_2 {x_i}^2$$

$$\mathbf{M3}: \mu_i = \alpha + \beta_1 x_i + \beta_2 {x_i}^2 + \\ + \beta_3 {x_i}^3 + \beta_4 {x_i}^4 + \beta_5 {x_i}^5$$

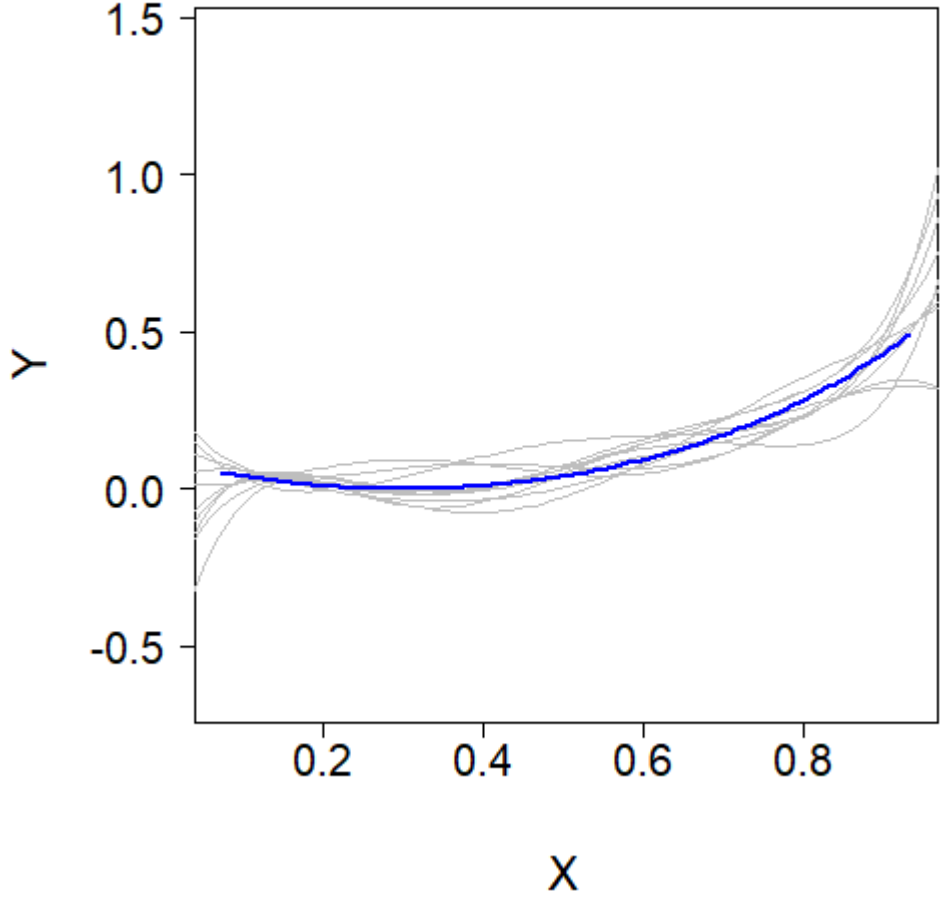


Instead of collecting data, we will simulate

```
# Define the dataset as a list for building
# our model in rethinking
set.seed(4221)
N <- 21
x <- runif(N)
d <- list(
  x = x,
  y = dexp((x - 0.3) ^ 2) - 1
)
```

Over-fitting

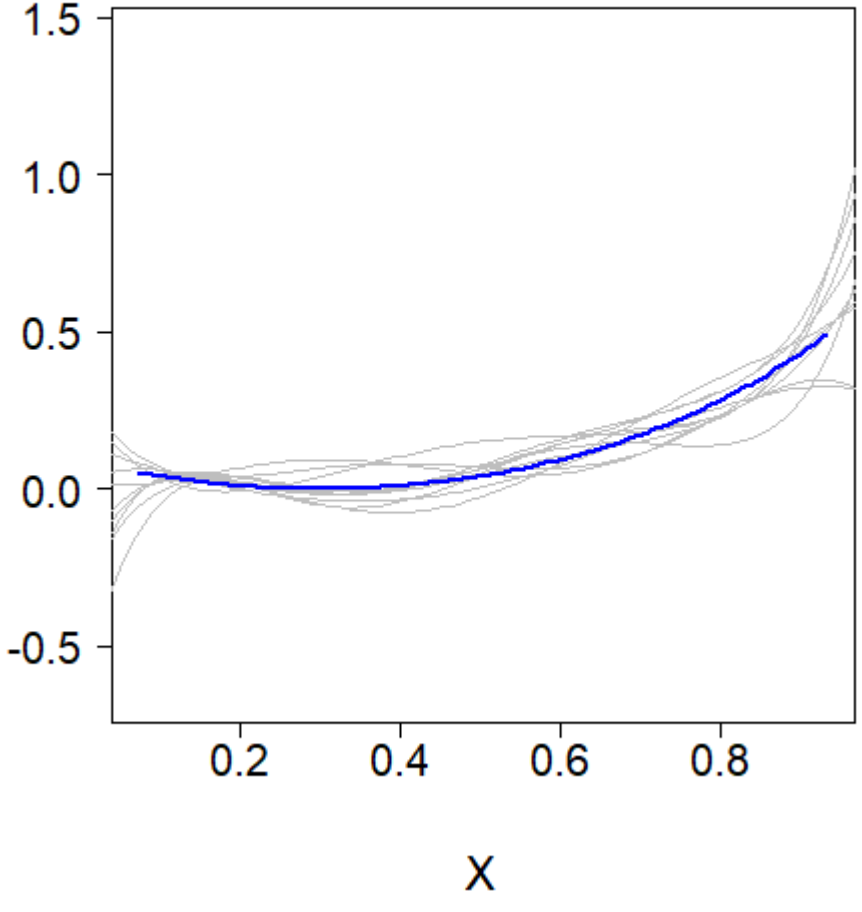
```
# Model 3: Higher-order polynomial model
m3 <- alist(
    y ~ dnorm(mu, sigma),
    mu <- a + b1 * x + b2 * x^2 + b3 * x^3 + b4
    a ~ dnorm(0, 1),
    b1 ~ dnorm(0, .1),
    b2 ~ dnorm(0, .1),
    b3 ~ dnorm(0, .1),
    b4 ~ dnorm(0, .1),
    b5 ~ dnorm(0, .1),
    sigma ~ dexp(1)
  )

fit_m3 <- quap(
  m3,
  data = d
)
```



Over-fitting

```
# Model 3: Higher-order polynomial model
precis(fit_m3)
```

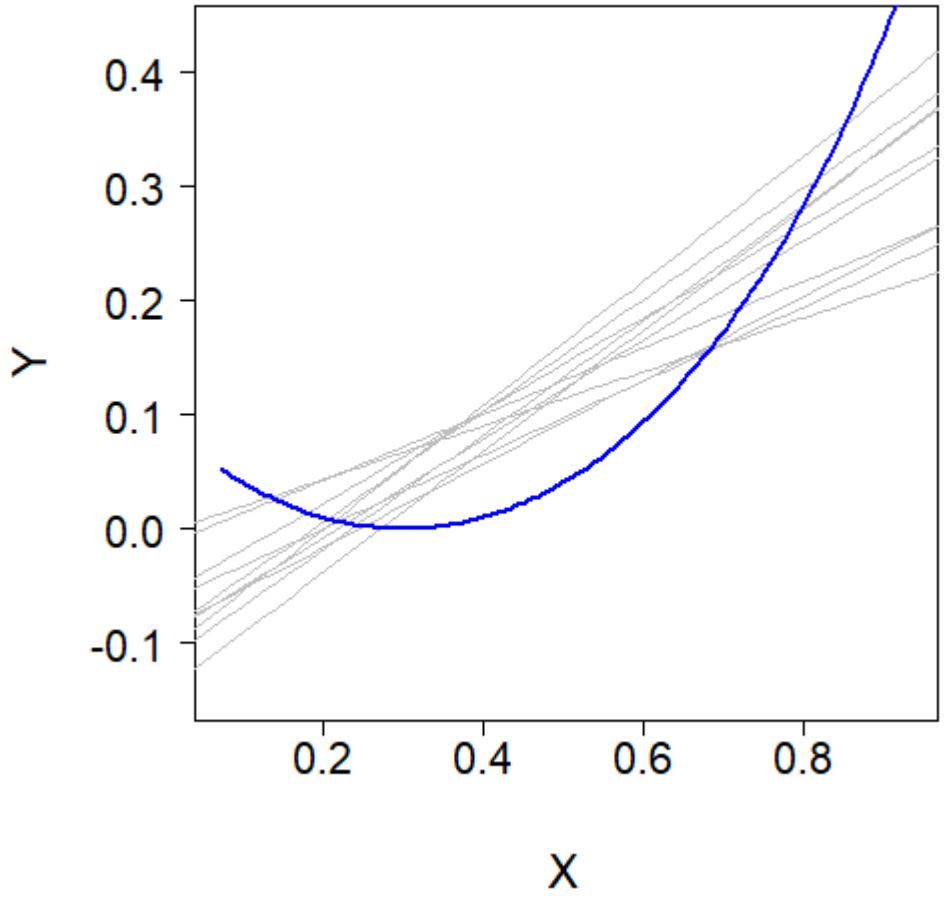
```
##               mean           sd         5.5%        94.5%
## a     -0.04049016 0.012729980 -0.06083513 -0.02014519
## b1     0.19672444 0.084275515  0.06203589  0.33141299
## b2    -0.26591334 0.107472122 -0.43767455 -0.09415214
## b3    -0.25959957 0.099705227 -0.41894778 -0.10025136
## b4    -0.10613584 0.082677361 -0.23827023  0.02599855
## b5     0.05116587 0.099429707 -0.10774200  0.21007375
## sigma  0.01064068 0.003260175  0.00543029  0.01585107
```



> To make inferences about unique features of the data at hand, as if they applied to all (or most all) samples (hence the population)
>
> Burham & Anderson (2002)

Under-fitting

```
# Model 1: Linear model
m1 <- alist(
    y ~ dnorm(mu, sigma),
    mu <- a + b * x,
    a ~ dnorm(0, 1),
    b ~ dnorm(0, 1),
    sigma ~ dexp(1)
  )
fit_m1 <- quap(
  m1,
  data = d
)
```



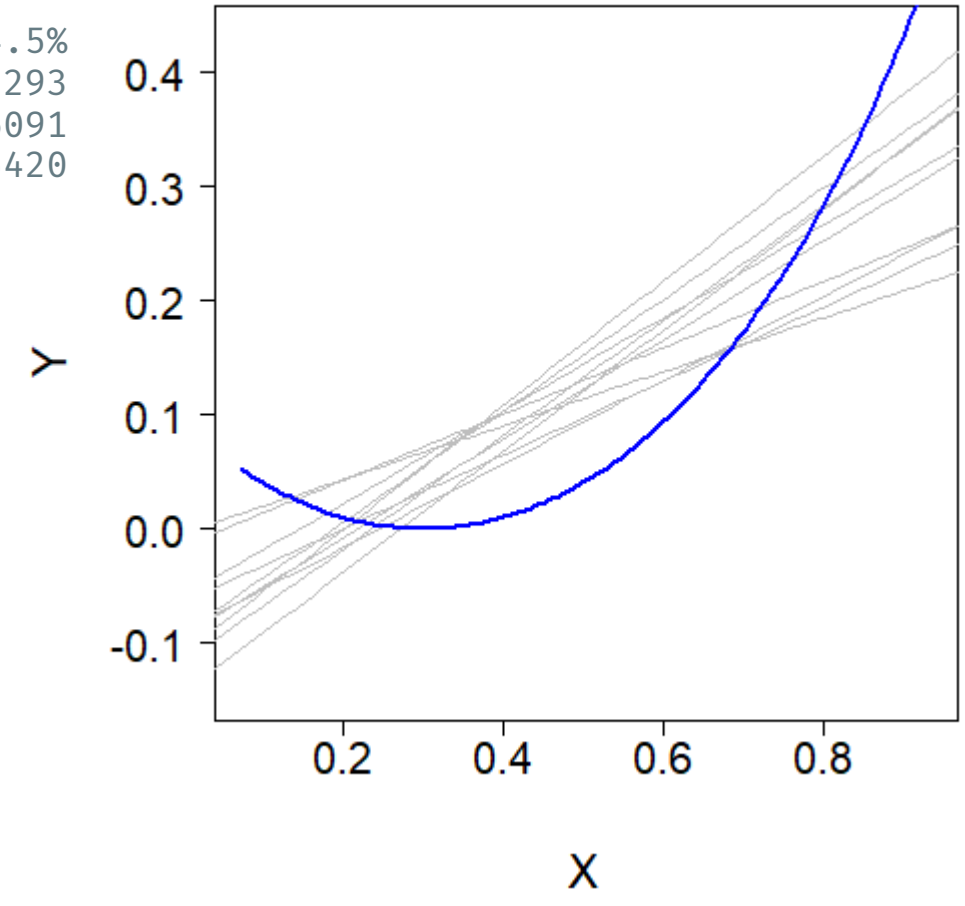
Under-fitting

```
precis(fit_m1)
```

```
##               mean           sd         5.5%        94.5%
## a      0.05225982 0.019436389  0.02119672  0.08332293
## b     -0.30053901 0.038216969 -0.36161710 -0.23946091
## sigma  0.04603736 0.007081021  0.03472052  0.05735420
```

> Failure to identify features in the data-generating process that are strongly replicable

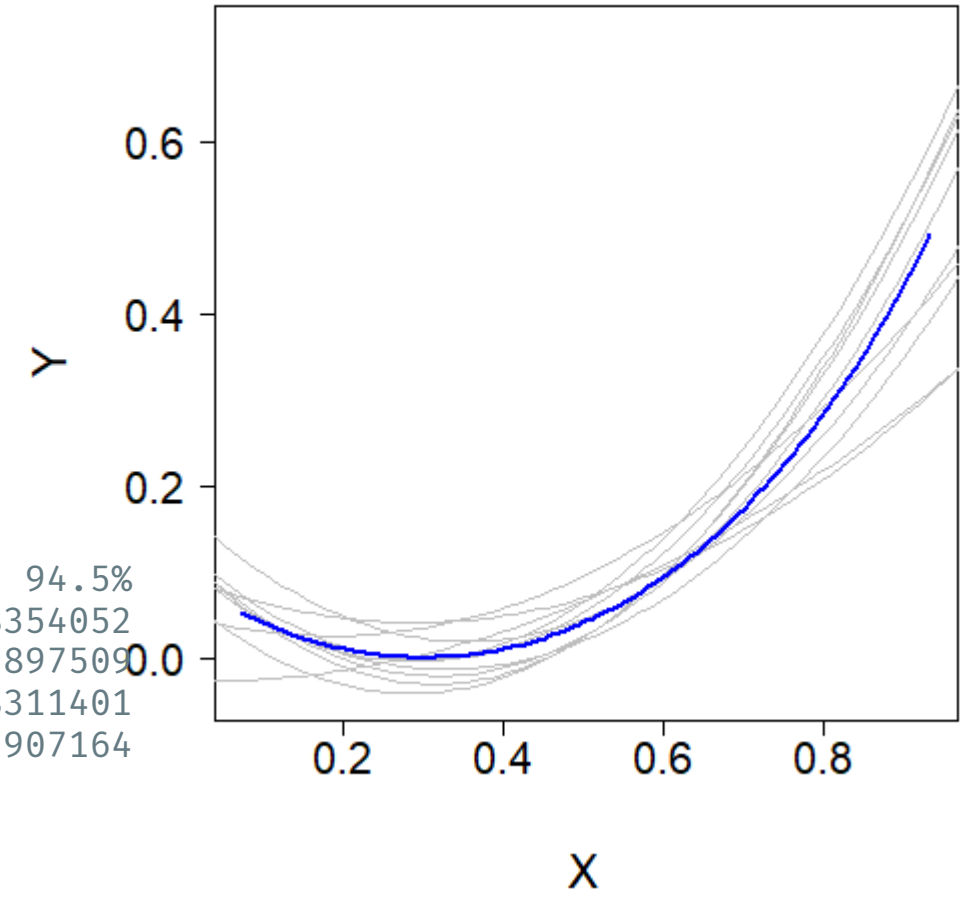
Burham & Anderson (2002)



A good fit

```
# Model 2: Quadratic model
m2 <- alist(
  y ~ dnorm(mu, sigma),
  mu <- a + b1 * x + b2 * x^2,
  a ~ dnorm(0, 1),
  b1 ~ dnorm(0, 1),
  b2 ~ dnorm(0, 1),
  sigma ~ dexp(1)
)
fit_m2 <- quap(
  m2,
  data = d
)
precis(fit_m2)
```

```
##                mean           sd         5.5%         94.5%
## a     -0.074127804 0.003612674 -0.079901556 -0.068354052
## b1     0.476156869 0.018608915  0.446416229  0.505897509
## b2    -0.818888308 0.019132172 -0.849465215 -0.788311401
## sigma  0.004906003 0.000626433  0.003904842  0.005907164
```



How to compare different model fit?

Mean Square Errors (MSE)

Mean Squared Error (MSE) is a measure of how well a model's predictions match the observed data. It is calculated as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Lower MSE indicates that predictions are closer to actual values
- Sensitive to outliers: Squaring the residuals means that large errors impact the MSE more
- Does not account for model complexity: A more complex model will always have lower MSE, even if it overfits

Rethinking MSE

$$\text{MSE}_{\text{Bayesian}} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbb{E}[\hat{y}_i])^2$$

```
calc_MSE_sim <- function(fit, data, response_var) {
  pred <- sim(fit, data = data) # Simulate new observations from the posterior
  pred_mean <- apply(pred, 2, mean) # Compute mean of simulated observations
  residuals <- data[[response_var]] - pred_mean
  mean(residuals^2)
}

calc_MSE_sim(fit_m1, d, "y")
```

```
## [1] 0.002072995
```

```
calc_MSE_sim(fit_m2, d, "y")
```

```
## [1] 2.377269e-05
```

```
calc_MSE_sim(fit_m3, d, "y")
```

Leave-One-Out Cross-Validation (LOO-CV)

- LOO-CV is a method to estimate a model's out-of-sample predictive accuracy

- It works by:

1. Removing one observation from the dataset
2. Fitting the model to the remaining data
3. Predicting the left-out observation and computing the error
4. Repeating this for every observation

- The average prediction error across all iterations gives a robust measure of model performance

Why use LOO-CV?

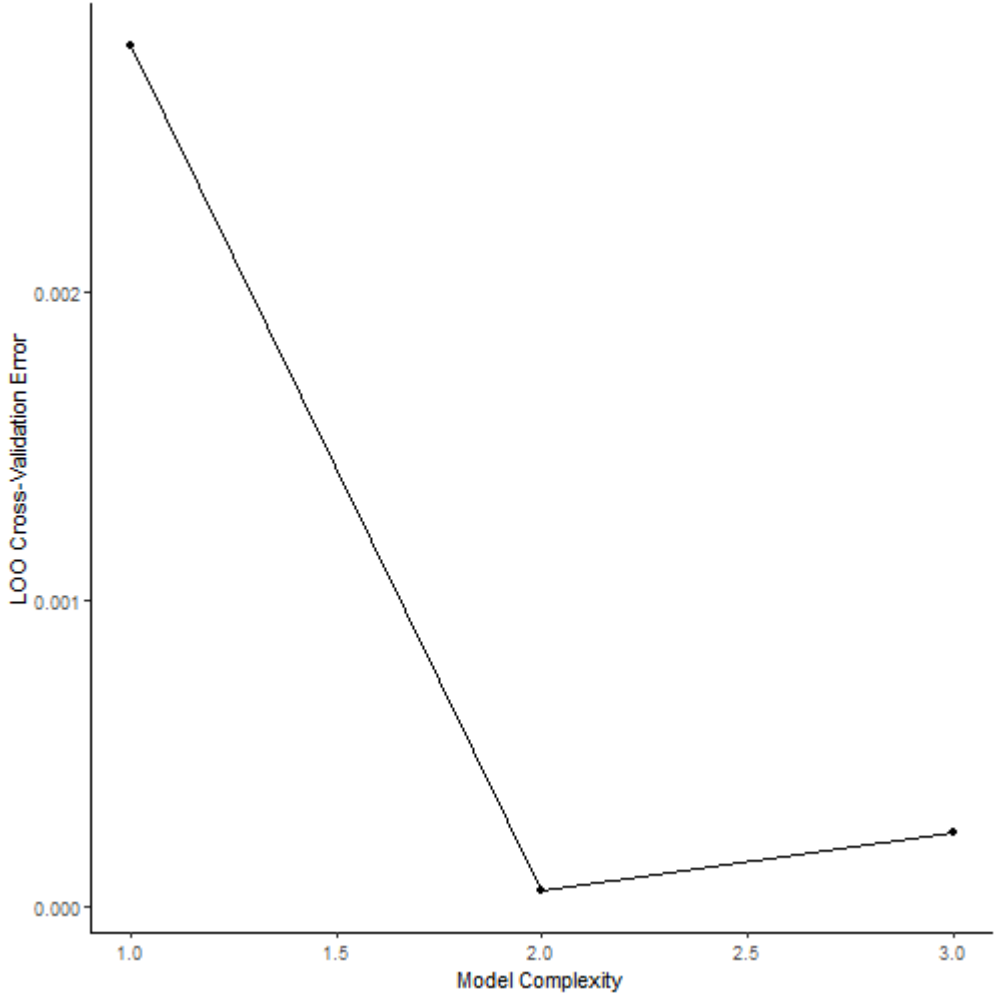
- Avoids overfitting – Complex models might fit training data well but fail on new data
- Better than Mean Squared Error (MSE) – MSE always decreases as model complexity increases, even when overfitting occurs
- More robust for Bayesian models – Bayesian LOO-CV accounts for uncertainty in parameter estimates.

Compute LOO-CV for our models

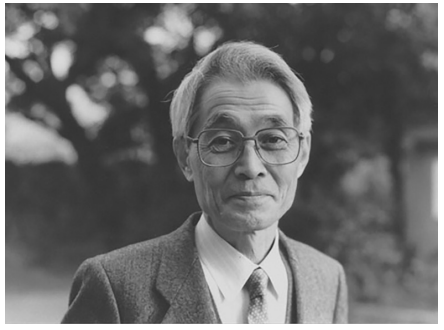
```
loo_cv = function(m, fit_m) {
  r = numeric(length(d$y)) # Store residuals

  for (i in 1:length(d$y)) {
    d_i = list(x = d$x[-i], y = d$y[-i])  # Remove one observation
    fit = quap(m, data = d_i, start = fit_m@start)  # Fit model without i-th obs
    # Predict for left-out observation using the posterior mean
    pred <- link(fit, data = list(x = d$x[i]))
    pred_mean <- mean(pred)  # Compute the mean prediction
    r[i] <- (d$y[i] - pred_mean)^2  # Squared error
  }

  mean(r)  # Return LOO-CV mean squared error
}
```

Compute LOO-CV for our models

```
models <- list(m1, m2, m3)
fit_models <-  list(fit_m1, fit_m2, fit_m3)
loo_errors <- map2_dbl(models, fit_models, loo
```



Akaike information criterion



Hirotugu Akaike
(1927-2009)

AIC estimates a statistical distance between two probability distributions: the true one, which is the reference, and a given model fitted to a sample from the true distribution.

AIC is an estimate of the information of the true data-generating distribution that is preserved by a model.

More specifically, AIC is an estimate of the Kullback-Leibler divergence (or K-L relative entropy) of the model to the reference true distribution that generated the data.

AIC for Gaussian models

$$AIC = -2 \sum_i \log[p(y_i|\theta)] + 2k$$

Where $\sum_i \log[p(y_i|\theta)]$ is the estimate of the residual variance of the model, in this case, the mean sum of squares of residuals:

$$\sum_i \log[p(y_i|\theta)] = \frac{1}{N} \sum_{i=1}^{N} (Y_i - \widehat{Y_i})^2$$

And K is the number of parameters of the model.

How to use AIC

- AIC expresses distance to the true model, or loss of information by the fitted model;
- Thus, the model with the lowest value of AIC among a set of competing models is the most plausible one (or best supported by the data);
- Canonical rule: models that differ ≤ 2 in their AIC values are equally supported by data;
- To ease model selection, we calculate $\Delta\mathrm{AIC}$:

$$\Delta_i = \mathrm{AIC}_i - \min(\mathrm{AIC})$$

- The best supported , or more plausible, model will have $\Delta_i = 0$

AICc: correction for small samples

For $n/K < 40$, multiply K by the correction term

$$\left(\frac{n}{n-K-1}\right)$$

AICc for Gaussian models:

$$\mathrm{AICc} = -2\sum_i \log[p(y_i|\theta)] + 2\,K\left(\frac{n}{n-K-1}\right)$$

Where n is the sample size.

Evidence weights

$$w_i = \frac{e^{-1/2\Delta_i}}{\sum e^{-1/2\Delta_i}}$$

- Evidence or Akaike weights sum up one;
- Thus, w_i express the relative support of each model in the set of competing models, in a standardized scale;
- In a frequentist approach, w_i estimates the probability that each model will be the best supported one, if you repeat the sample and then the selection many times
- In a Bayesian framework, w_i are interpreted as a relative measure of model plausibility based on their out-of-sample predictive accuracy, i.e. they reflect predictive performance, not absolute model probability

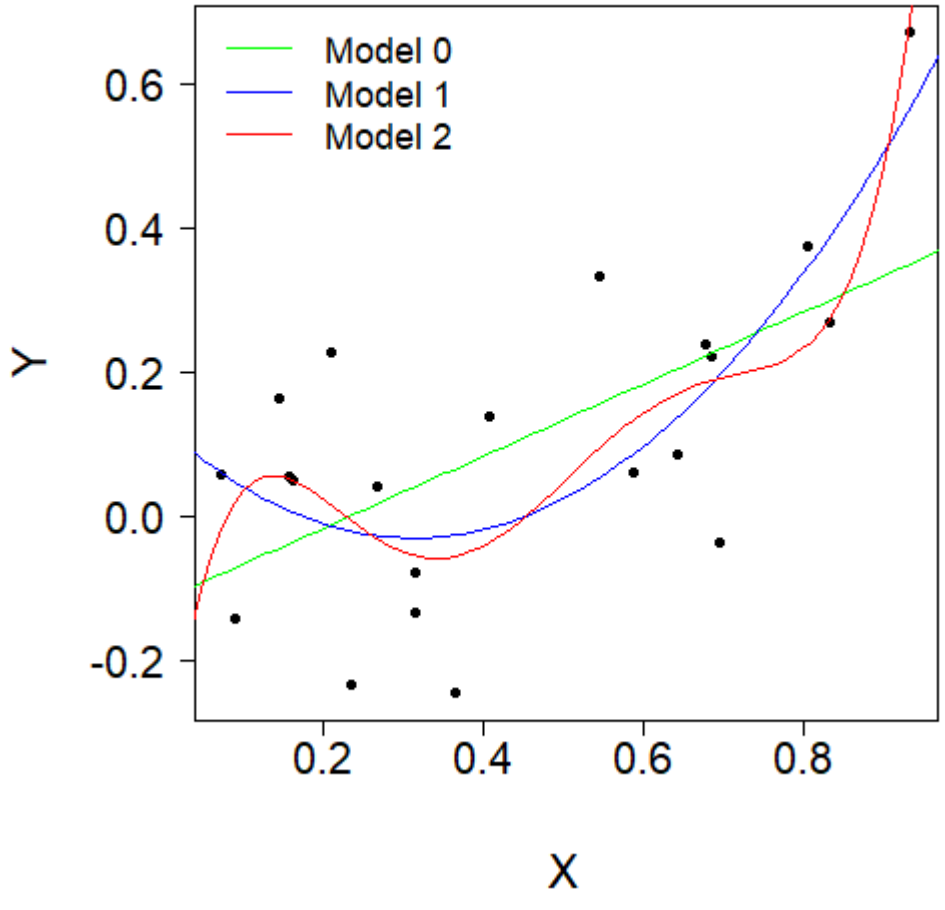
Model selection with AIC: example

```
library(bbmle)
AICctab(fit_m1, fit_m2, fit_m3, logLik = TRUE,
        weights = TRUE, nobs = N)
```

```
##        logLik AICc   dLogLik dAICc  df weight
## fit_m2   82.0 -153.5   47.2     0.0 4   1
## fit_m3   65.8 -108.9   30.9    44.7 7   <0.001
## fit_m1   34.8  -62.3    0.0    91.3 3   <0.001
```

```
AICtab(fit_m1, fit_m2, fit_m3, logLik = TRUE, I
       weights = TRUE, nobs = N)
```

```
##        logLik AIC    dLogLik dAIC   df weight
## fit_m2   82.0 -156.0   47.2     0.0 4   1
## fit_m3   65.8 -117.5   30.9    38.5 7   <0.001
## fit_m1   34.8  -63.7    0.0    92.4 3   <0.001
```



Remarks on AIC

- Ties can and do happen: more than one model with $\Delta_i < 2$ tell us that the data does not contain enough evidence to spot the best model;
- Model selection with AIC is not a statistical test;
- Model selection is restricted to competing models: if all competing models are bad, the selected model will just be the least bad.
- AIC does not express goodness of fit. The selected model can still have a poor fit;
- AIC cannot be used to compare models fitted to different datasets;
- For the same reason, the AIC cannot be used to compare models fitted to transformed and untransformed data.

Incorporating uncertainty into model comparisons

Widely Applicable Information Criterion or Watanabe-Akaike Information Criterion (WAIC)

Sumio Watanabe

- WAIC (Widely Applicable Information Criterion) estimates out-of-sample predictive accuracy
- Like AIC, it penalizes model complexity, but incorporates posterior uncertainty
- The key concept is the log pointwise predictive density (lppd), which replaces the likelihood in AIC

Define Log Pointwise Predictive Density (lppd)

$$\text{lppd} = \sum_i \log \left(\frac{1}{S} \sum_{s=1}^{S} p(y_i|\theta_s) \right)$$

- Instead of using a single maximum likelihood estimate, WAIC averages log-likelihoods over posterior samples (\theta_s)
- This means parameter uncertainty is incorporated directly

Accounting for Model Complexity

$$\text{WAIC} = -2 \left[\text{lppd} - \sum_i \text{var}_\theta \log p(y_i|\theta) \right]$$

- Penalty for model complexity: The second term is the variance of log-likelihood across posterior samples
- Higher variance = More model flexibility = Stronger penalty to prevent overfitting

WAIC formula & interpretation

$$\text{WAIC} = -2 \sum_i \log \left(\frac{1}{S} \sum_{s=1}^{S} p(y_i|\theta_s) \right) + 2p_{\text{WAIC}}$$

where:

- S = number of posterior samples
- $theta_s$ = posterior sample s
- p_{WAIC} = $\sum_i \text{var}_\theta \log p(y_i|\theta)$ (effective number of parameters)

Why Use WAIC?

- Incorporates parameter uncertainty (unlike AIC)
- Works for hierarchical & complex Bayesian models
- Useful for model comparison – lower WAIC = better predictive accuracy

Back to our three models

$$\mathbf{M1}: \mu_i = \alpha + \beta_1 x_i$$

$$\mathbf{M2}: \mu_i = \alpha + \beta_1 x_i + \beta_2 {x_i}^2$$

$$\mathbf{M3}: \mu_i = \alpha + \beta_1 x_i + \beta_2 {x_i}^2 + \\ + \beta_3 {x_i}^3 + \beta_4 {x_i}^4 + \beta_5 {x_i}^5$$

```
# Computes WAIC-based model comparison
compare(fit_m1, fit_m2, fit_m3, func = WAIC)
```

```
##              WAIC        SE    dWAIC       dSE     pWAIC       weight
## fit_m2 -149.67873 13.726260  0.00000        NA  7.603714 1.000000e+00
## fit_m3 -102.63255 16.404528 47.04618 14.276198 16.391116 6.082082e-11
## fit_m1  -59.70291  8.717573 89.97581  7.627784  5.003421 2.897344e-20
```

Comparison of Model Selection Methods

Criterion	What it Measures	Complexity Penalty	Parameter Uncertainty	Predictive Accuracy	Computational Cost	Best Use Case
MSE	Squared error between predictions & observed data	No	No	No	Low	Quick model fit check
AIC	Model fit via likelihood & complexity tradeoff	Yes	No	No	Low	Frequentist model selection
LOO-CV	Out-of-sample predictive accuracy	Yes	Yes	Yes	High	Bayesian model comparison
WAIC	Bayesian model comparison balancing fit & complexity	Yes	Yes	Yes	Moderate	Fully Bayesian alternative to AIC

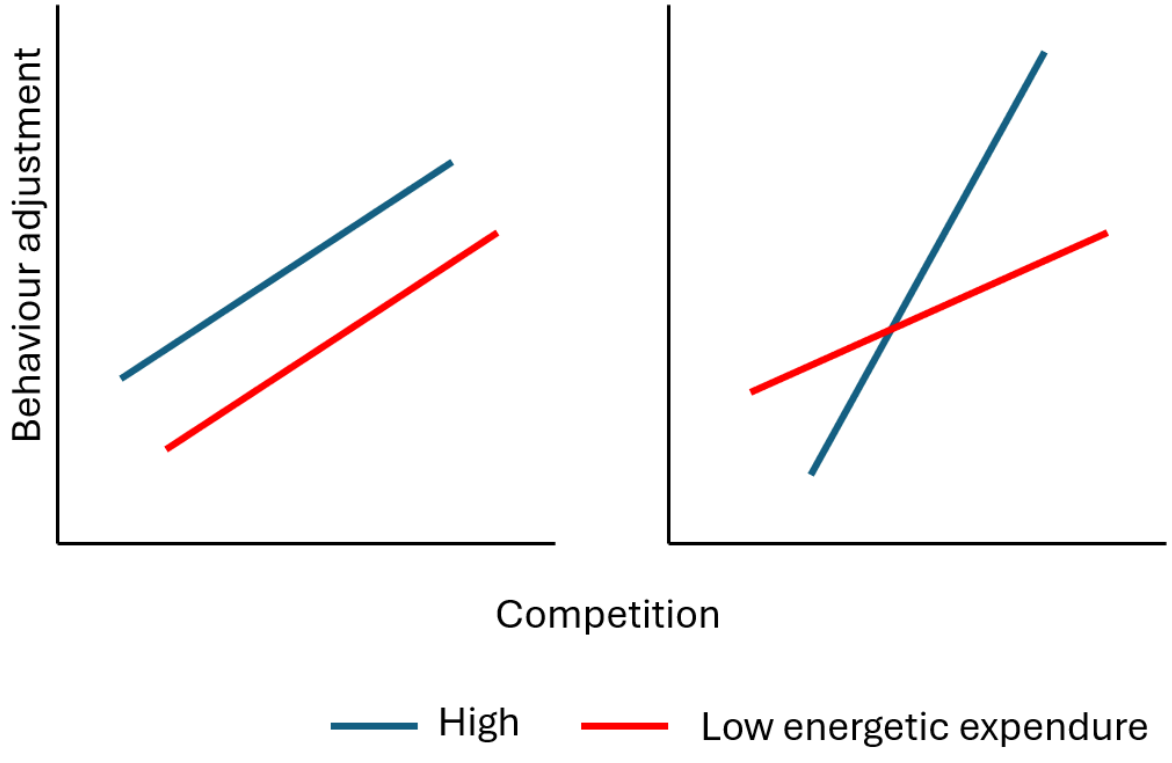
The Ecological Detective: Field Course 2024

Atlantic Forest field course

- The antlion larvae adopts a sit-and-wait predatory strategy by constructing funnel-shaped traps in the soil
- This study investigates how energy expenditure and intraspecific competition affect trap adjustment behavior and body condition
- They expect a positive relationship between adjustment behavior and interespecific competition
- They also expect that depending on the environment the energetic demand would be different

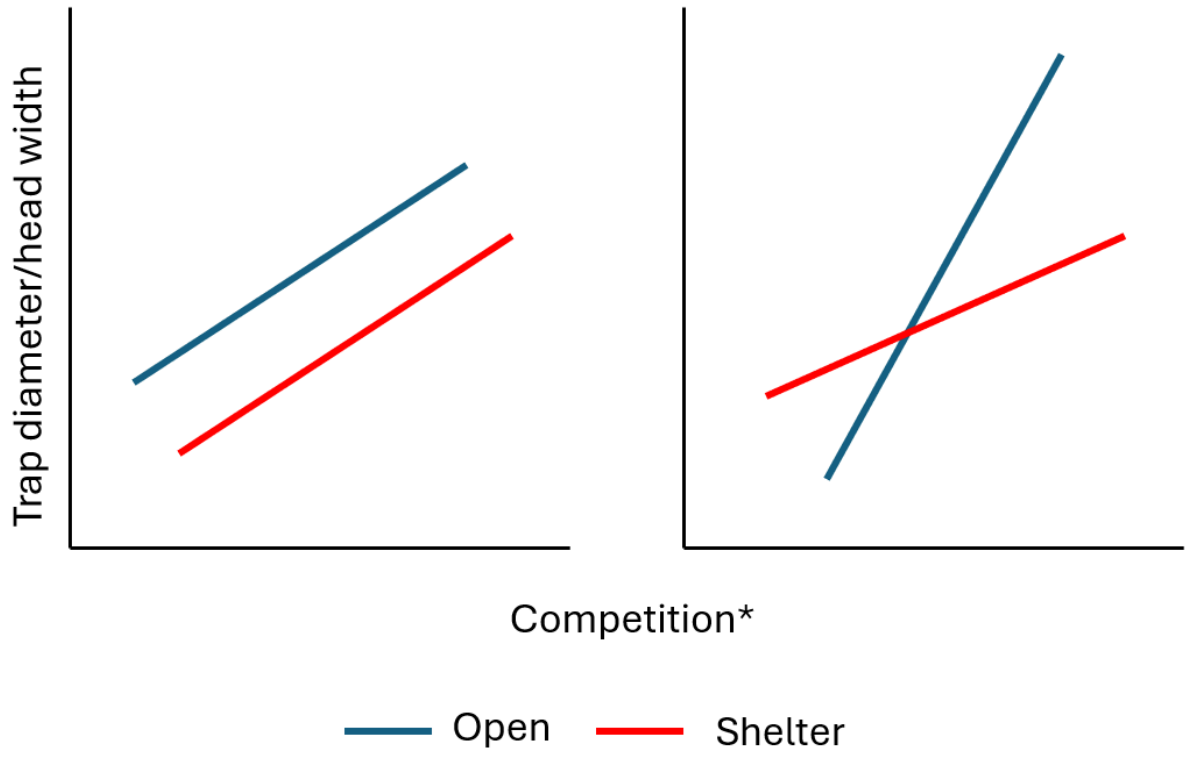
Hypotheses

The more intense the **interespecific competition**, the more the antlion needs to adjust its **behavior**, and in **high demanding environments**, this relationship is even stronger



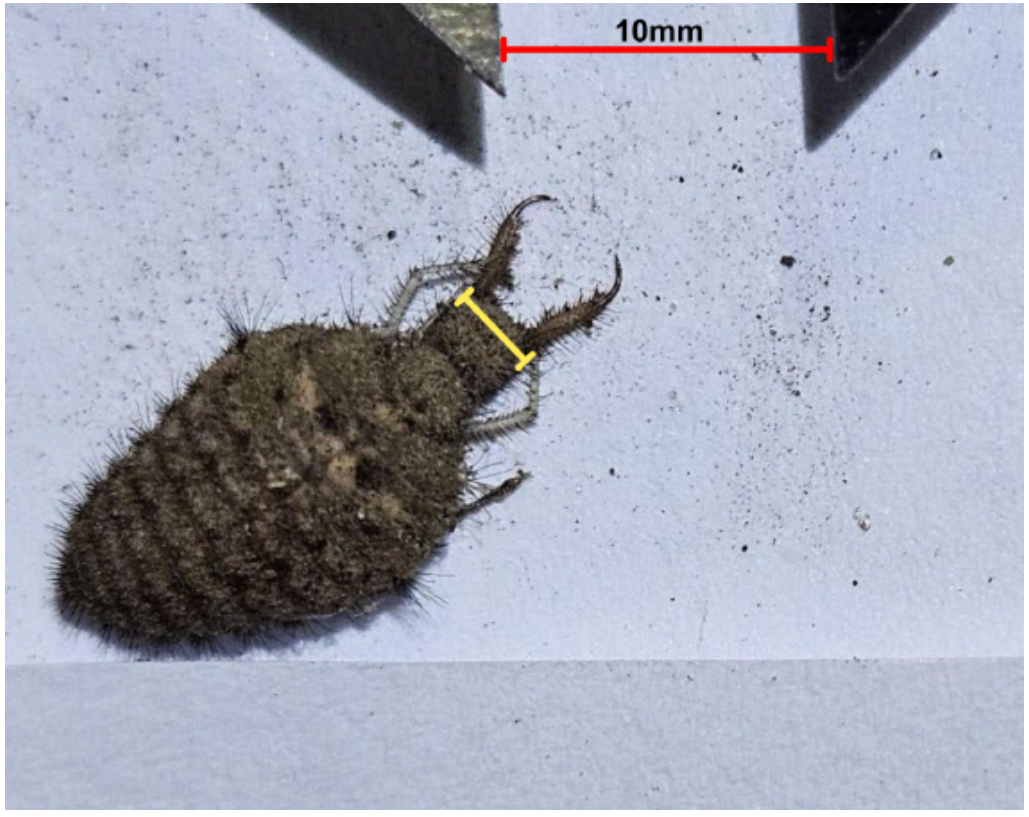
Predictions

The more intense the interespecific competition, the higher the ratio $trap\ diameter/head\ width$, and in exposed environments, this relationship is even stronger, i.e. changes in α only or both α and β

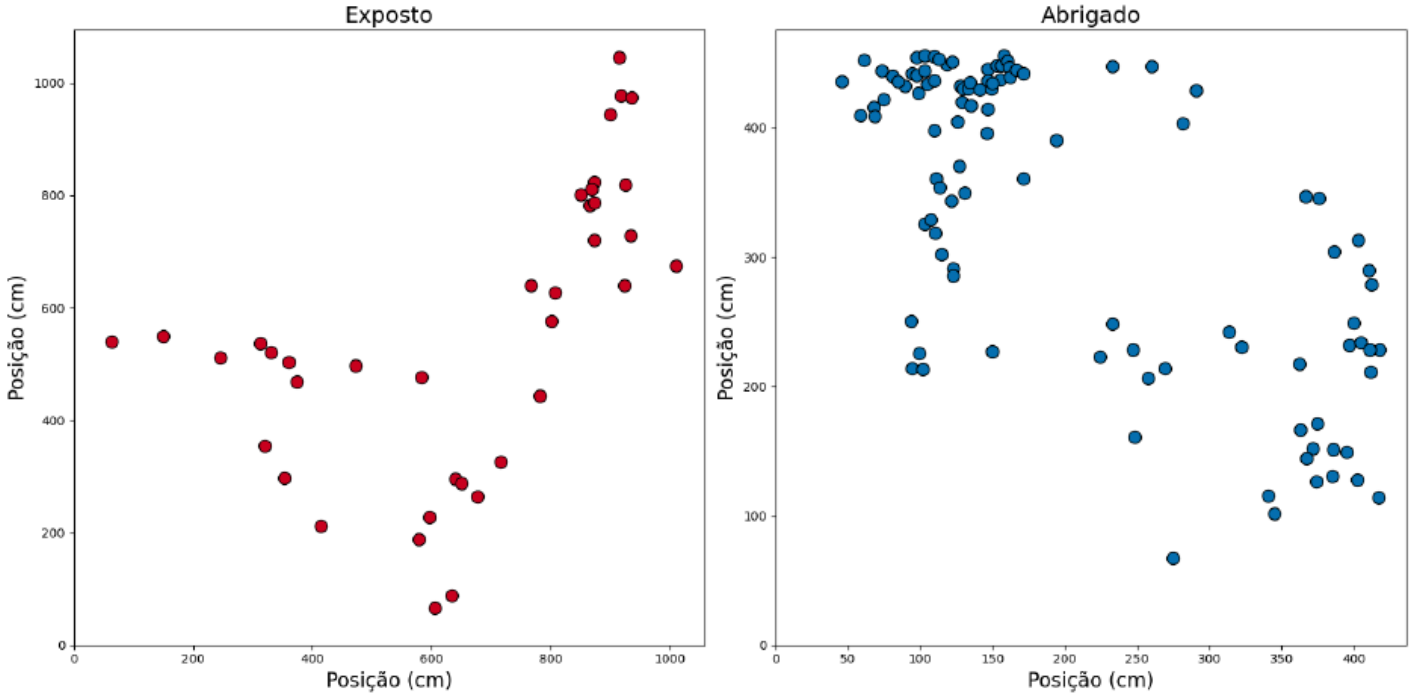


The environment

The behaviour adjustment



Understanding the interespecific competition



Competition variable

$$Competition_i = \frac{1}{K} \sum_{j=1}^{k} \frac{1}{D_{ij}}$$

- K = number of competitors, maximum 3
- D_{ij} = distance between individual i and the competitor j

Translating the hypothesis into models

$$Y_i \sim Normal(\mu, \sigma)$$

1. $\mu_i = \alpha + \beta * Environment + \beta * C$
2. $\mu_i = \alpha_{Env} + \beta * Environment + \beta * C$
3. $\mu_i = \alpha + \beta * C$
4. $\mu_i = \alpha + \beta * Environment$
5. $\mu_i = \alpha$

The data

```
head(data)
```

```
##    Env        C    ratio
## 1 open 1.506429 30.10252
## 2 open 1.397413 25.78991
## 3 open 1.465836 22.10678
## 4 open 1.416630 17.03102
## 5 open 1.792986 37.64292
## 6 open 2.013705 37.73293
```

```
data$Env_fac <- as.numeric(as.factor(data$Env) == "shelter")
data$Env_int <- as.numeric(as.factor(data$Env))
```

Model #1

$$\mu_i = \alpha + \beta * Environment + \beta * C$$

```
m1 <- alist(
  ratio ~ dnorm(mu, sigma),
  mu <- a + b1*Env_fac + b2*C,
  a ~ dnorm(20, 5),
  b1 ~ dnorm(0, 1),
  b2 ~ dnorm(0, 1),
  sigma ~ dexp(1)
)
```

Model #2

$$\mu_i = \alpha_{Env} + \beta * Environment + \beta * C$$

```
m2 <- alist(
  ratio ~ dnorm(mu, sigma),
  mu <- a[Env_int] + b1*Env_fac + b2*C,
  a[Env_int] ~ dnorm(20, 5),
  b1 ~ dnorm(0, 1),
  b2 ~ dnorm(0, 1),
  sigma ~ dexp(1)
)
```

Model #3

$$\mu_i = \alpha + \beta * C$$

```
m3 <- alist(
  ratio ~ dnorm(mu, sigma),
  mu <- a + b*C,
  a ~ dnorm(20, 5),
  b ~ dnorm(0, 1),
  sigma ~ dexp(1)
)
```

Model #4

$$\mu_i = \alpha + \beta * Environment$$

```
m4 <- alist(
  ratio ~ dnorm(mu, sigma),
  mu <- a + b*Env_fac,
  a ~ dnorm(20, 5),
  b ~ dnorm(0, 1),
  sigma ~ dexp(1)
)
```

Model #5

$$\mu_i = \alpha$$

```
m5 <- alist(
  ratio ~ dnorm(mu, sigma),
  mu <- a,
  a ~ dnorm(20, 5),
  sigma ~ dexp(1)
)
```

Model fit

```
m1_fit <- quap(m1, data = data)
m2_fit <- quap(m2, data = data)
m3_fit <- quap(m3, data = data)
m4_fit <- quap(m4, data = data)
m5_fit <- quap(m5, data = data)
```

Model comparison

```
compare(m1_fit, m2_fit, m3_fit, m4_fit, m5_fit, func = WAIC)
```

```
##            WAIC       SE    dWAIC      dSE    pWAIC     weight
## m2_fit 749.8972 19.91900 0.000000       NA 4.106904 0.73733197
## m1_fit 753.2311 18.26691 3.333931 4.349914 3.243604 0.13922240
## m3_fit 755.2390 17.74252 5.341748 5.684025 2.821182 0.05101726
## m4_fit 755.6789 17.67361 5.781733 5.333986 2.678392 0.04094262
## m5_fit 756.2042 17.21700 6.307006 6.126250 2.300551 0.03148576
```

Comparing with posterior distributions

Step #1: Simulate posterior distributions

- The `extract.samples()` function generates posterior samples for the model parameters
- Useful for understanding uncertainty in parameters estimates

```
# Number of posterior draws
n_samples <- 1000

# Extract posterior samples
posterior <- extract.samples(m2_fit, n = n_samples)
```

Comparing with posterior distributions

Step #2: Generate predictions for each C value

```r
# Create a sequence of C values for prediction
C_seq <- seq(from = min(data$C), to = max(data$C), length.out = 100)

# Matrices to store predictions
mu_matrix_open <- matrix(NA, nrow = n_samples, ncol = length(C_seq))  # For "open"
mu_matrix_shelter <- matrix(NA, nrow = n_samples, ncol = length(C_seq))  # For "shelter"

# Loop over posterior samples
for (i in 1:n_samples) {
  # Compute mu for each C_seq
  mu_matrix_open[i, ] <- posterior$a[i, 1] + 0 * posterior$b1[i] + posterior$b2[i] * C_seq  # "Open
  mu_matrix_shelter[i, ] <- posterior$a[i, 2] + 1 * posterior$b1[i] + posterior$b2[i] * C_seq  # "Sh
}

# Compute mean and credible intervals
mu_mean_open <- apply(mu_matrix_open, 2, mean, na.rm = TRUE)
mu_PI_open <- apply(mu_matrix_open, 2, function(x) PI(x[is.finite(x)], prob = 0.89))

mu_mean_shelter <- apply(mu_matrix_shelter, 2, mean, na.rm = TRUE)
mu_PI_shelter <- apply(mu_matrix_shelter, 2, function(x) PI(x[is.finite(x)], prob = 0.89))  # Apply
```

Comparing with posterior distributions

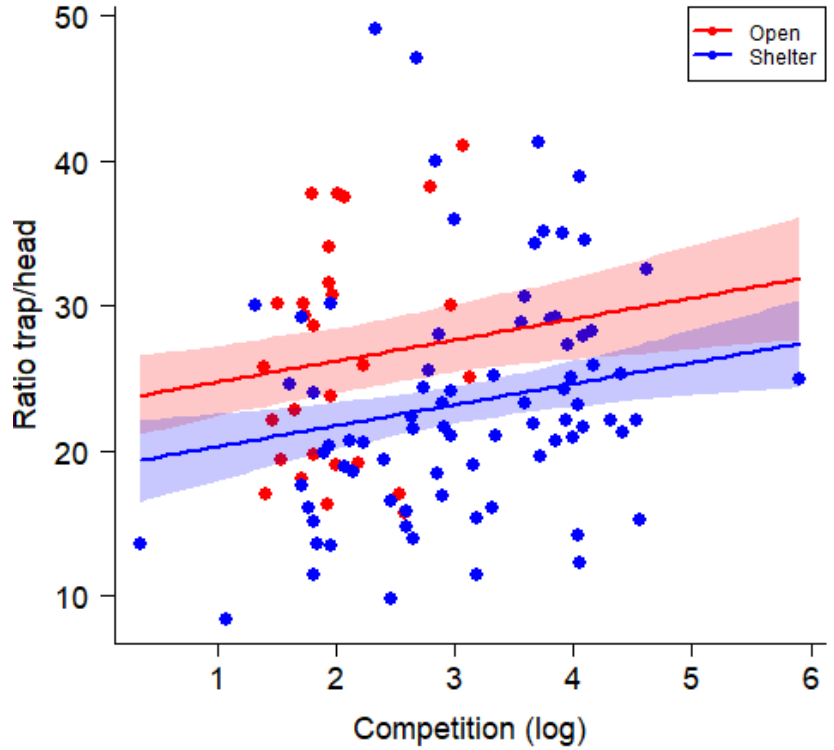
Step #4: Simulate new observations

- The `sim()` function generates posterior predictive distributions
- Samples from the full posterior including the observation model (likelihood)
- Useful for simulating new observations
- Predict new data based on posterior uncertainty

```
# Generating posterior samples
posterior_sim <- sim(m2_fit, n = 100)
```

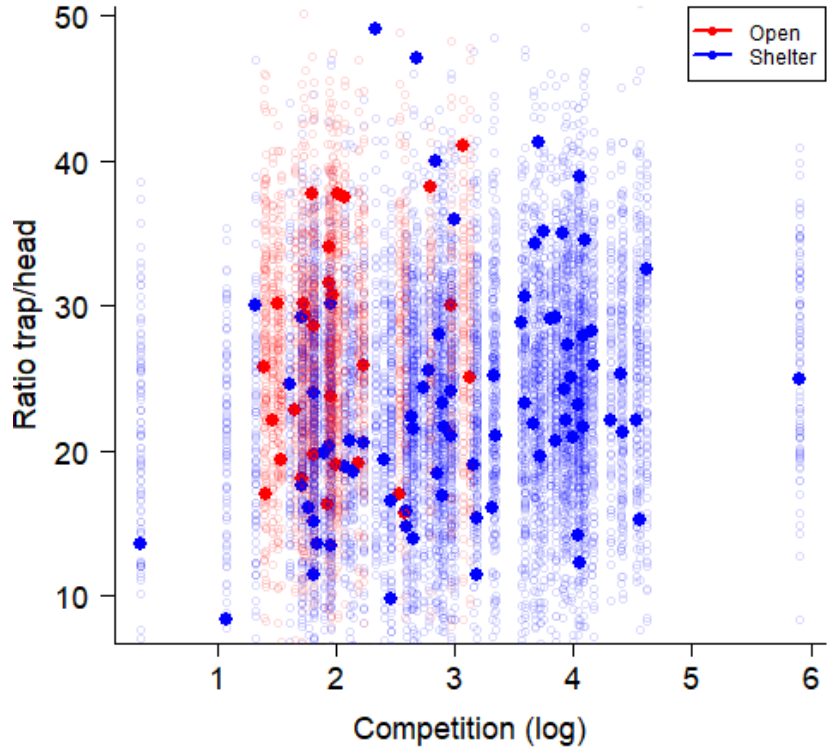
Comparing with posterior distributions

Step #4: Plot the results

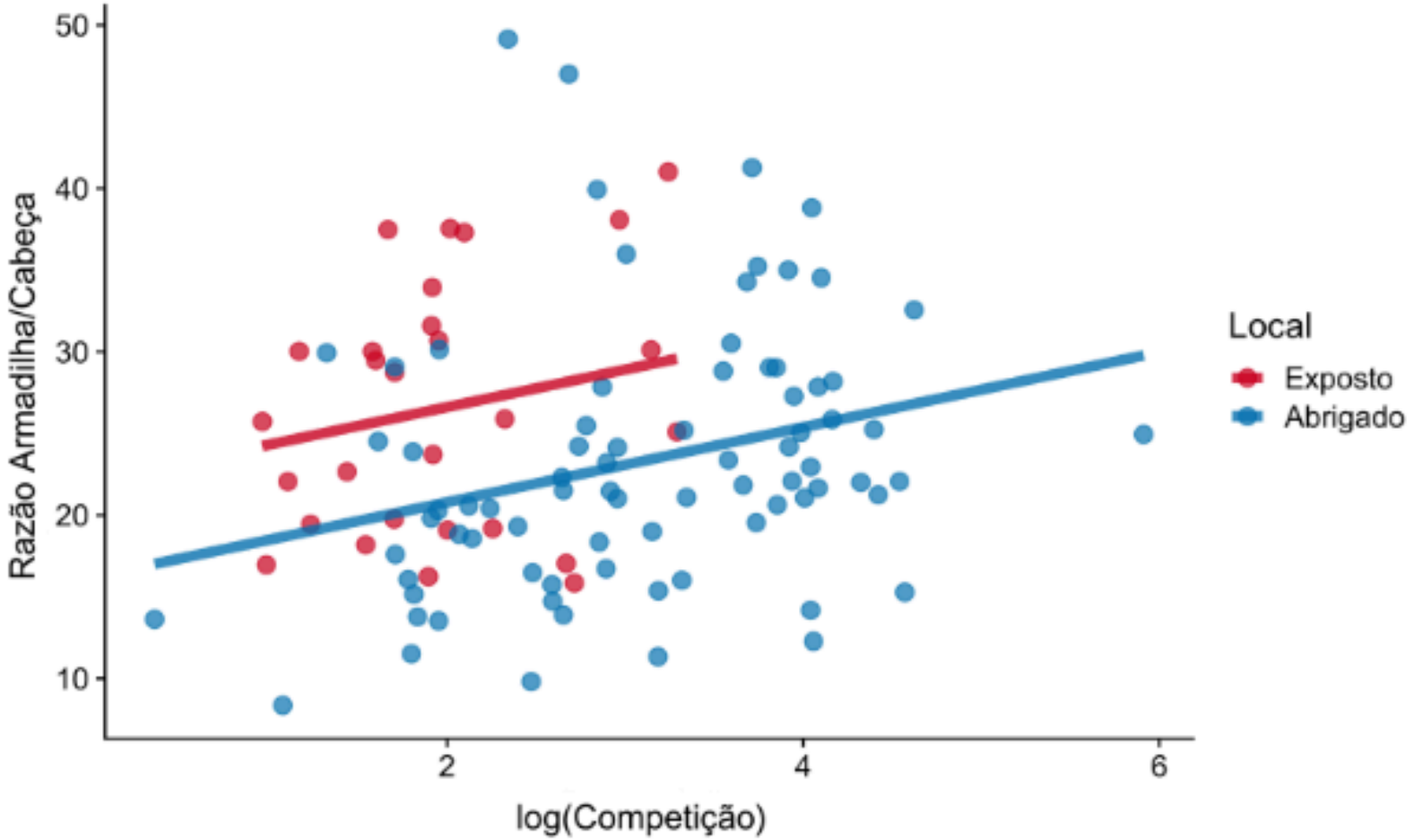


Comparing with posterior distributions

Step #4: Plot the results



What they found



Model Selection in the Bayesian Context

Bayesian model selection is diverse: Multiple criteria exist depending on the goal of inference

Bayesian model selection differs from frequentist approaches by incorporating uncertainty in model comparison

Unlike AIC, Bayesian model selection methods often rely on posterior model probabilities, computed using:

- Bayes Factors – Relative support for models via marginal likelihoods
- WAIC – Fully Bayesian alternative to AIC, penalizing model complexity using posterior variance
- Cross-validation (LOO-CV) – Estimates predictive performance by testing model predictions on held-out data
- No single best approach – Choice depends on whether the goal is prediction, explanation, or inference

Hooten and Hobbs 2015

Key Takeaways

- MSE: Measures fit but does not help with model selection
- AIC: Penalizes complexity but ignores parameter uncertainty
- LOO-CV: Best for prediction, but computationally expensive - PSIS-LOO (Pareto Smoothed Importance Sampling LOO) provides a computationally efficient approximation to traditional LOO-CV
- WAIC: A fully Bayesian alternative to AIC that incorporates uncertainty

Which method to use?

✅ If you want prediction, use LOO-CV.

✅ If you want Bayesian model comparison, use WAIC.

✅ If you are working in frequentist settings, use AIC.

Further reading

- Burnham, K. P., & Anderson, D. R. 2002. Model Selection and Multimodel Inference: A Practical-Theoretic Approach, 2nd ed. New York, Springer-Verlag.
- Bolker, B.M. 2008. Ecological Models and Data in R. Princeton: Princeton University Press.
- Hilborn R, Mangel M. 2013. The ecological detective: confronting models with data (MPB-28). Princeton University Press; 2013 Dec 31.
- Hooten, M.B. and Hobbs, N.T. 2015. A guide to Bayesian model selection for ecologists. Ecological monographs, 85(1), pp.3-28.
- McElreath, R., 2018. Statistical Rethinking: A Bayesian Course with Examples in R and Stan, 2nd ed. CRC Press.