Course Review

Michael Noonan

Biol 520C: Statistical modelling for biological data

Table of contents

Design- vs. Model-based Inference

The Scientific Method

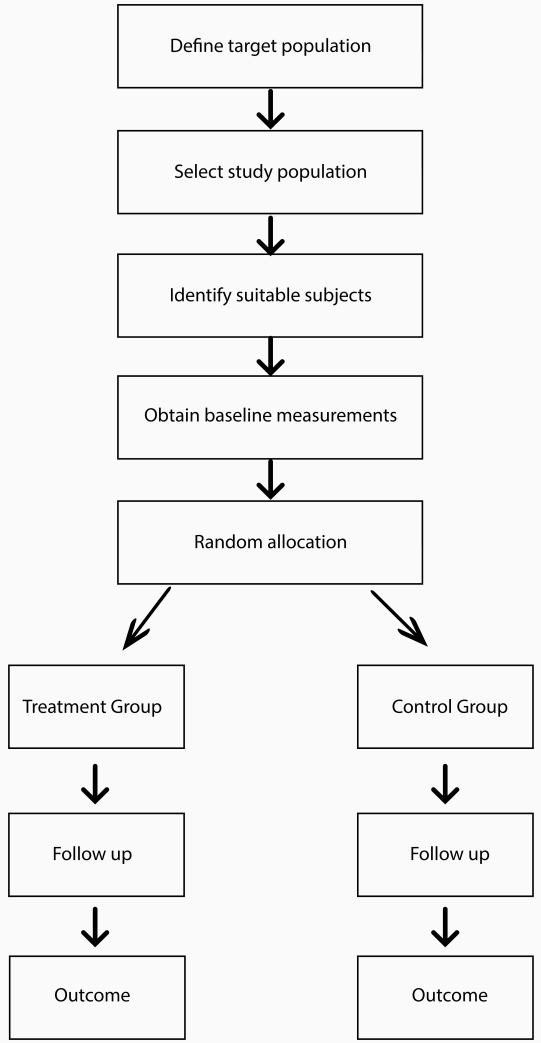
Science is a process of learning about nature. As scientists, we weigh competing ideas about how the world works (hypotheses) against observations (data).

But our descriptions of the world are almost always incomplete, our observations have error, and important data is often missing... So we use statistics to confront our observe with our hypotheses.

The process of making scientific inference can be split into two broad categories: design-based and model-based

Design-based inference

- In design-based inference, most of the focus is on experimental design
- Inference is focused towards a target population
- Data are typically analysed by comparing means and variances across groups (e.g., ANOVAs, *t*-tests, etc...)



Limitations of design-based inference

Many biological processes have long time-scales.

Many biological systems have very poor reproducibility.

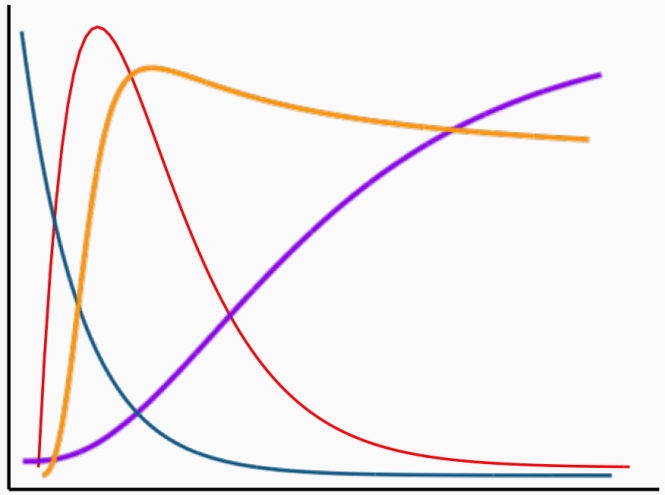
Experiments in biological systems often result in ambiguous outcomes.

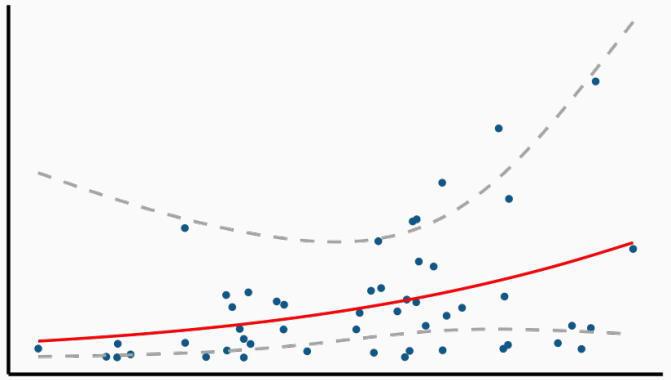
Biological data regularly break key assumptions for conventional tests.

Controlled experiment are difficult to carry out in the wild.

Model-based inference

- In model-based inference, most of the focus is on identifying an unknown deterministic model.
- Inference can be extrapolated beyond the target population.
- In model-based inference you make distributional assumptions to make your response a random variable.
- Data are typically analysed by fitting a model to data and interpreting the parameter estimates.





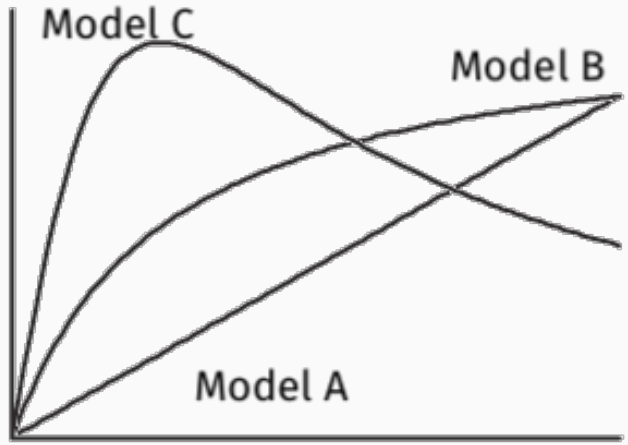
One hypothesis, many models

A single hypothesis can be represented by multiple models.

Hypothesis: Body mass M increases with age L

Models:

- $M = aL$ Model A: Body mass is proportional to age
- $M = \frac{AL}{1+bL}$ Model B: Body mass saturates as age increases
- $M = aLe^{-bL}$ Model C: Body mass increases and then decreases as age increases

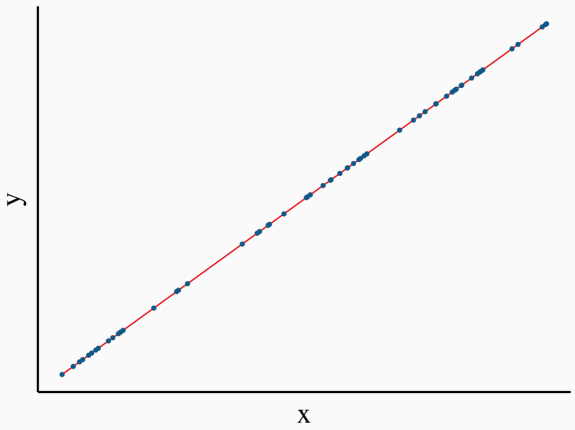


Source: Hillborn and Mangel 1997

Deterministic vs. Stochastic

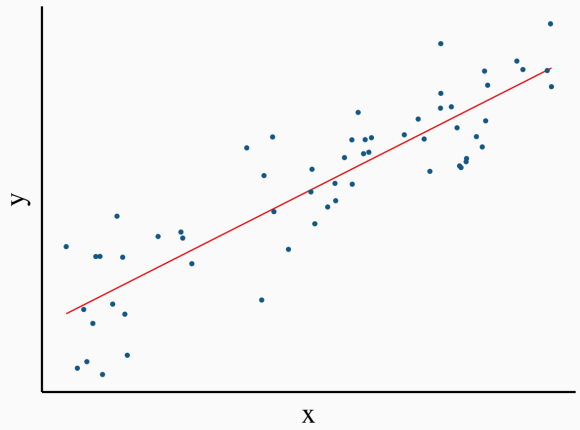
Deterministic models

- No components are uncertain
- Outcome is always the same
- $y_i = \beta_0 + \beta_1 x_i$



Stochastic models

- Some components are uncertain and characterised by probability distributions
- Outcome is variable
- $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$



Linear Regression

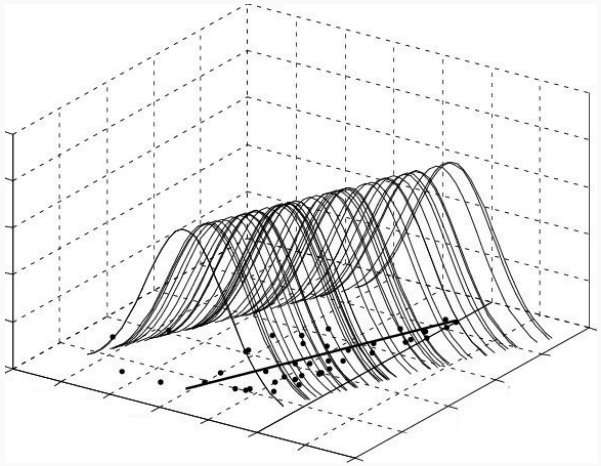
Linear regression assumptions

Applying standard linear regression to a problem relies on satisfying 5 assumptions:

- Correct model specification
- Normality of the residuals
- Homogeneity
- Fixed x
- Independence

Residuals as diagnostic tools

Residuals are what's left in your data after your model has done its work.



Source: Zuur et al. 2009

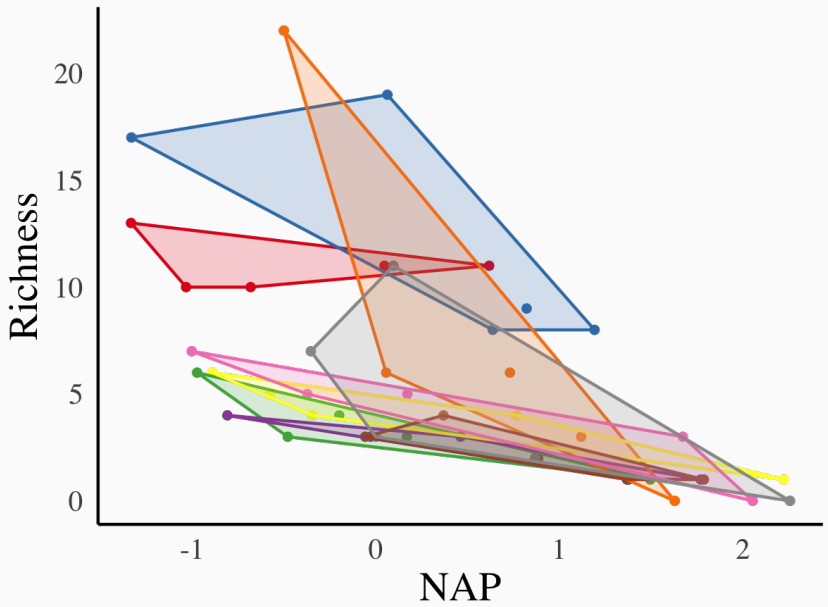
Fitting models of the general form: $Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2)$, the residuals should have a very specific structure.

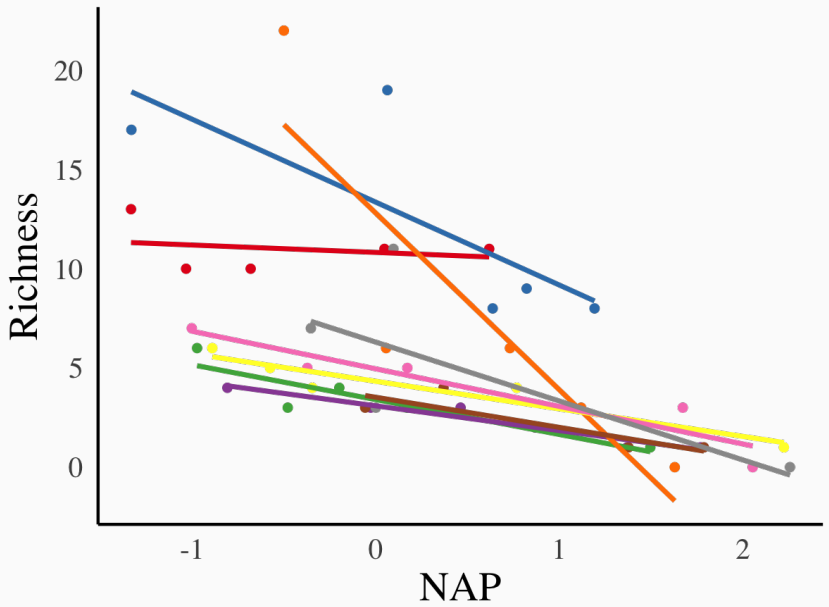
By definition, if these models are behaving properly they should result in some amount of residual spread around values predicted by a model's deterministic component and deviations from this expectation provide you with clues on how you might be able to improve the fit of your model.

Mixed Effects Models

Nested Data

We then saw how nested data can impart structure to a dataset that needs to be modelled, and we did this using mixed effects models.





In matrix notation a linear mixed effects model can be represented as

$$\boldsymbol{y_i} = X_i\boldsymbol{\beta} + Z_i\boldsymbol{b_i} + \boldsymbol{\varepsilon_i}$$

$\boldsymbol{y_i}$ is the vector of observations ($N \times 1$ vector);

X_i is a matrix of our 'fixed' predictor variables ($N \times p$ matrix);

$\boldsymbol{\beta}$ is a vector of fixed effects ($p \times 1$ vector);

Z_i is a matrix of our random predictor variables ($N \times qJ$ matrix for q random effects and J groups);

$\boldsymbol{b_i}$ is a vector of random effects $\sim \mathcal{N}(0, G_i)$ ($qJ \times 1$ vector);

$\boldsymbol{\varepsilon_i}$ is our distribution of errors $\sim \mathcal{N}(0, \sigma_i)$.

$$\underbrace{\mathbf{y}}_{\text{N x 1}} = \overbrace{\underbrace{\mathbf{X}}_{\text{N x p}} \quad \underbrace{\boldsymbol{\beta}}_{\text{p x 1}}}^{\text{N x 1}} + \overbrace{\underbrace{\mathbf{Z}}_{\text{N x qJ}} \quad \underbrace{\boldsymbol{b}}_{\text{qJ x 1}}}^{\text{N x 1}} + \underbrace{\boldsymbol{\varepsilon}}_{\text{N x 1}}$$

Linear Regression

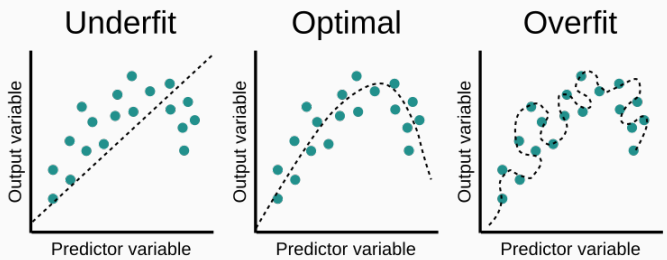
formula	meaning
`(1|group)`	random group intercept
`(x|group)` = `(1+x|group)`	random slope of x within group with correlated intercept
`(0+x|group)` = `(-1+x|group)`	random slope of x within group: no variation in intercept
`(1|group) + (0+x|group)`	uncorrelated random intercept and random slope within group
`(1|site/block)` = `(1|site)+(1|site:block)`	intercept varying among sites and among blocks within sites (nested random effects)
`site+(1|site:block)`	*fixed* effect of sites plus random variation in intercept among blocks within sites
`(x|site/block)` = `(x|site)+(x|site:block)` = `(1 + x|site)+(1+x|site:block)`	slope and intercept varying among sites and among blocks within sites
`(x1|site)+(x2|block)`	two different effects, varying at different levels
`x*site+(x|site:block)`	fixed effect variation of slope and intercept varying among sites and random variation of slope and intercept among blocks within sites
`(1|group1)+(1|group2)`	intercept varying among crossed random effects (e.g. site, year)

source: https://bbolker.github.io/mixedmodels-misc/glmmFAQ.htmllinear-mixed-models

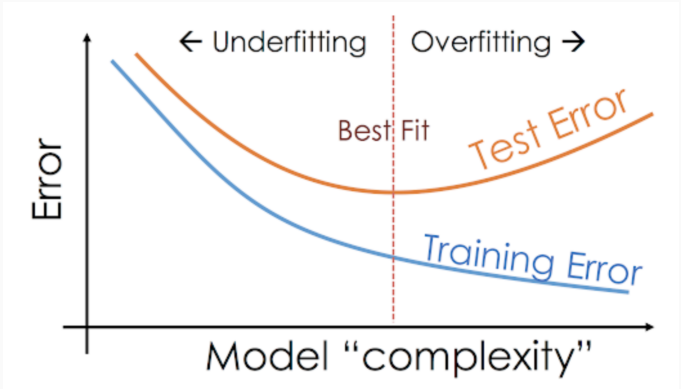
Overfitting and Model Selection

The Problem of Overfitting

An **underfit** model fails to accurately predict the data that were used to fit the model, and test datasets or future conditions.



An **overfit** model gives a very low prediction error on the dataset used to fit the model, but has a very high prediction error on test data.



This happens because you're fitting the noise not the signal.

Likelihood-Ratio Test

The likelihood-ratio test compares a pair of **nested** models based on the ratio of their likelihoods.

$$\lambda_{\mathsf{LR}} = -2\ln\left[\frac{\mathcal{L}(\text{Reduced model})}{\mathcal{L}(\text{Full model})}\right]$$

The likelihood-ratio test statistic is often expressed as a difference between the log-likelihoods

$$\lambda_{\mathsf{LR}} = -2(\ln[\mathcal{L}(\text{Reduced})] - \ln[\mathcal{L}(\text{Full})])$$

Accessible in `R` via the `anova()` function with the p-value being based off of the χ^2 distribution.

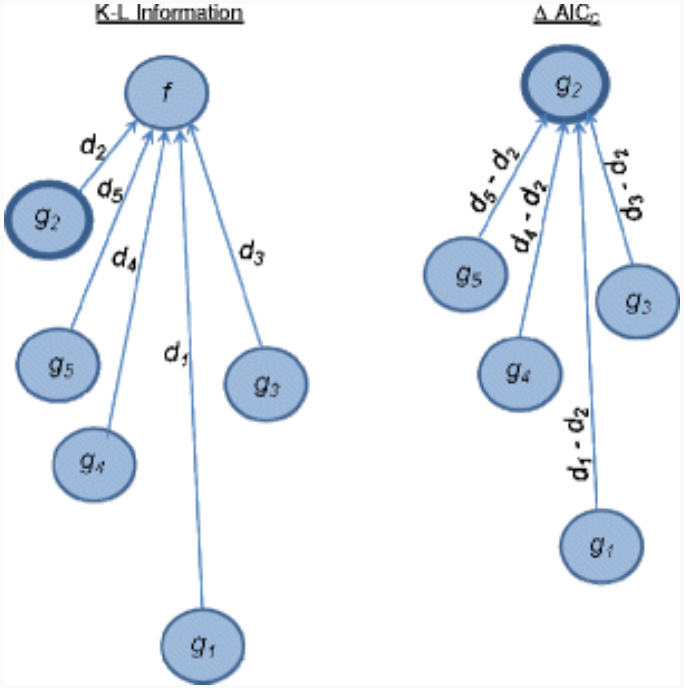
Akaike Information Criterion

K-L information represents the distance between model g_i and reality, but because we can't estimate 'reality', we instead rely on AIC (or other IC) and rank models amongst one another (the lower the value the better).

When the sample sizes are small, there is a good chance that AIC will overfit, so we prefer AICc in practice

$$\mathrm{AICc} = \mathrm{AIC} + \frac{2k^2+2k}{n-k-1}$$

There's also a formal Relationship between ΔAIC and LRTs: $\Delta\mathrm{AIC} = \lambda_{\mathrm{LR}} - 2(K_2 - K_1)$



(Burnham *et al.*, 2011)

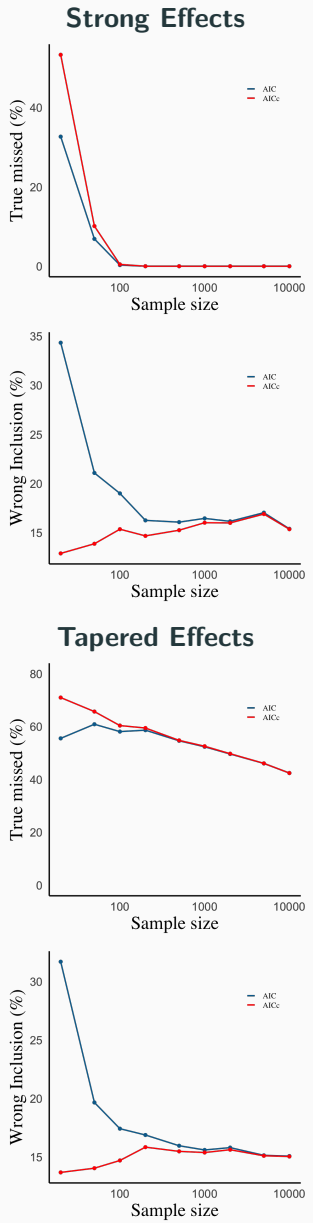
AIC/AICc performance

For systems with strong effect sizes, both AIC and AICc identified true parameters well for high n.

For systems with tapered effect sizes, neither AIC nor AICc identified all of the true parameters well even with n was extremely high.

When n was small, AIC missed fewer true parameters than AICc, but at the cost of more false positives.

For both systems AIC and AICc consistently identified noise parameters as being important.



Model averaging

Model averaging refers to the practice of using several models at once for making predictions or inferring parameters.

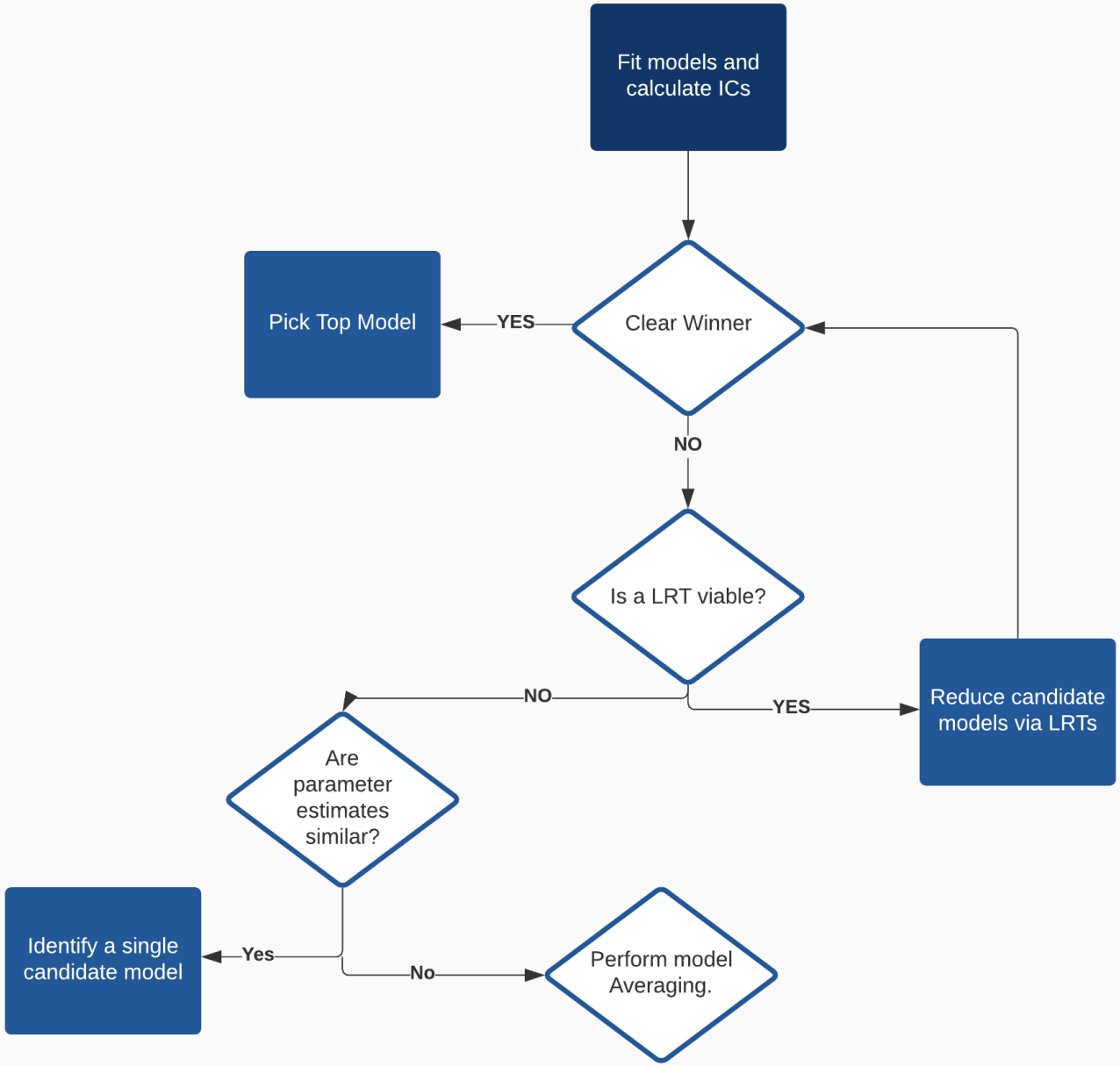
The Idea: If a model is misspecified, the parameters estimates may be too high/low averaging parameter values from different models, with biases in either way, should cancel out and reduce bias in the average.

Model averaging has no super-powers. Like most other statistical methods, model averaging has benefits and costs, and you must weight them to decide which approach is best for your problem.

Benefits include a possible reduction of predictive error and improved parameter estimates.

Costs include extra work/computation time, the fact that it does not always work, and that confidence intervals and p-values are difficult to provide.

Pragmatic workflow



Autocorrelation and Heteroskedasticity

Variance-Covariance Matrix

Correcting for autocorrelation and/or heteroskedasticity is 'simply' involves identifying the autocorrelation/ heteroskedasticity structure of the residuals and modifying the variance-covariance matrix.

$$V = \sigma^2 \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

Overview of Variance Structures

We covered several possible ways to model heteroskedastic data:

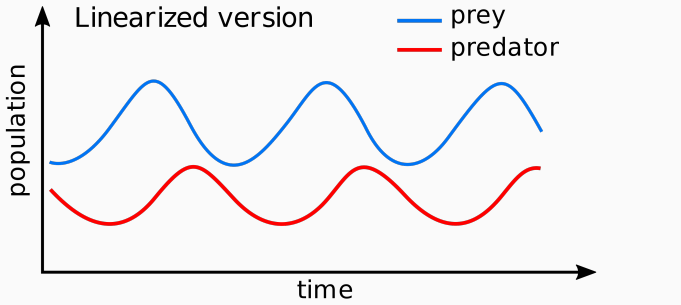
Type	**Formula**	**DF**	**R Function**
Fixed	$\varepsilon_i \sim \mathcal{N}(0, \sigma^2 \times x_i)$	0	`varFixed()`
Constant	$\varepsilon_{ij} \sim \mathcal{N}(0, \sigma_j^2)$	j-1	`varIdent()`
Power	$\varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2 \times \lvert x_i \rvert^{2\delta_j})$	1 or j	`varPower()`
Exponential	$\varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2 \times e^{2\delta_j \times x_i})$	1 or j	`varExp()`
Const.+Power	$\varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2 \times (\delta_{1j} + \lvert x_i \rvert^{\delta_{2j}})^2)$	1 or 2j	`varConstPower()`
Combination	Variable	Var.	`varComb()`

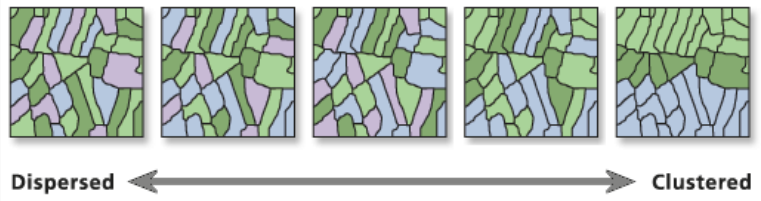
Sources of autocorrelation

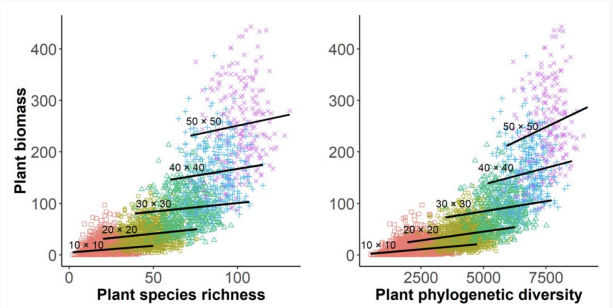
We then saw how anything that causes some data points to be more similar to each other than others can result in autocorrelation.

Over the course of three lectures we covered the three most common sources of autocorrelation in biological data:

- **Time**: Data that are close together in time are more related.
- **Space**: Data that are close together in space are more related.
- **Phylogeny**: Species that are closer together on an evolutionary timescale are more related.







(Liang *et al.*, 2019)

Autocorrelation impact

Sample size, n is the denominator when calculating both SEs and CIs.

$$\mathrm{SE} = \frac{\sigma}{\sqrt{n}} \qquad 95\%\mathrm{CI} = \bar{x} \pm 1.96\frac{\sigma}{\sqrt{n}}$$

All else equal: $\uparrow n = \downarrow \mathrm{SE}$ & $\downarrow \mathrm{CI}$

But with autocorrelated data each new datapoint is related to a previously collected datapoint and does not bring a full independent datapoint worth of information (e.g., 90% autocorr. $\approx$ 10% new info).

When data are autocorrelated $n_{\mathrm{effective}} < n$, meaning SEs and CIs shrink faster than they should, resulting in a false sense of confidence.

Effect is usually strongest on SEs and CIs, but autocorrelation can also impact the mean: $\bar{x} = \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) = \frac{x_1 + \cdots + x_n}{n}$

Temporal Autocorrelation Structures

We covered several ways to model temporally autocorrelated data:

Type	**Covariance ρ**	**DF**	**R Function**
IID	0	0	`corSymm()`
Compound Symetric	$\rho = \frac{\theta}{\theta+\sigma^2}$	1	`corCompSymm()`
AR-1	$\rho^{\lvert t-s \rvert}$	1	`corAR1()`
ARMA	variable	variable	`corARMA()`

Going from IID to AR-1 offered a big improvement, and then fine-tuning via more complicated ARMA structures resulted in only marginal improvements over AR-1. This is common in practice.

Unless there are serious issues remaining in your residuals, the pragmatic solution is to stop when you have a reasonably appropriate model.

Temporal Autocorrelation Structures

The autocorrelation functions require you to specify a formula for the autocorrelation.

When you specify `lme(..., correlation = corAR1())` this is equivalent to `lme(..., correlation = corAR1(form = ~1 | id))` and assumes that the measurements are equally spaced and in the correct order.

When you specify `lme(..., correlation = corAR1(form = ~time | id))` you use the time variable time to determine how far apart the measurements are, and define the time lag.

`corAR1()` works with discrete time. There is also `corCAR1()` that works with continuous time.

Spatial Autocorrelation Structures

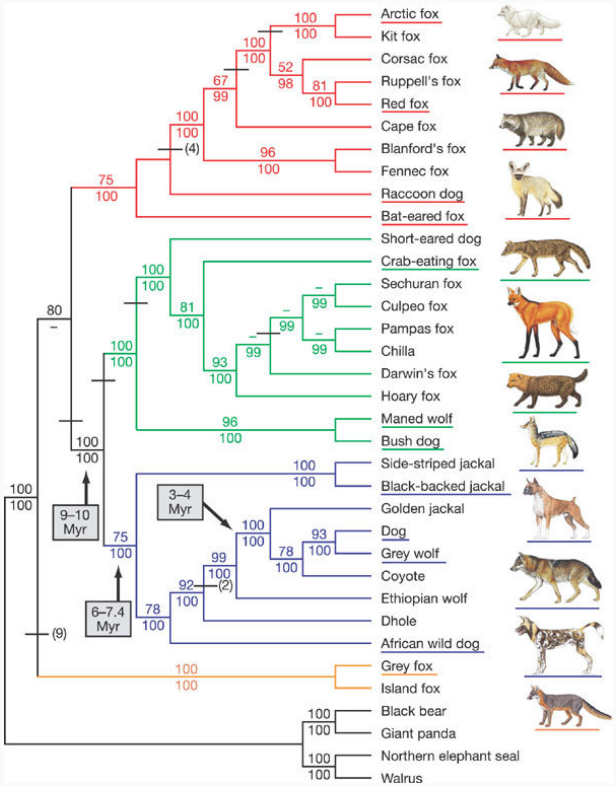
We covered several ways to model spatially autocorrelated data:

Type	Description	R Function
IID	0	`corSymm()`
Linear	$\Phi = 1 - (1\frac{D}{\rho})I(d < \rho)$	`corLin()`
Spherical	$\Phi = 1(1 - 1.5(\frac{d}{\rho}) + 0.5(\frac{d}{\rho})3)I(d < \rho)$	`corSpher()`
Gaussian	$\Phi = 1 - e^{-(\frac{D}{\rho})^2}$	`corGaus()`
Exponential	$\Phi = 1 - e^{-\frac{D}{\rho}}$	`corExp()`
Rational quadratic	$\Phi = \frac{1}{1+(\frac{\rho}{d})^2}$	`corRatio()`

The model structures can be difficult to interpret, but their variograms have very recognizable features. Familiarising yourself with them will help you quickly narrow down what structure to use.

Phylogenetic regression

We saw how we can use information from phylogenetic trees to build variance covariance matrices and correct for phylogenetic inertia



$$V = \begin{bmatrix} \sigma^2 & 0 & \cdots & 0 \\ 0 & \sigma^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma^2 \end{bmatrix}$$

Phylogenetic Corrections

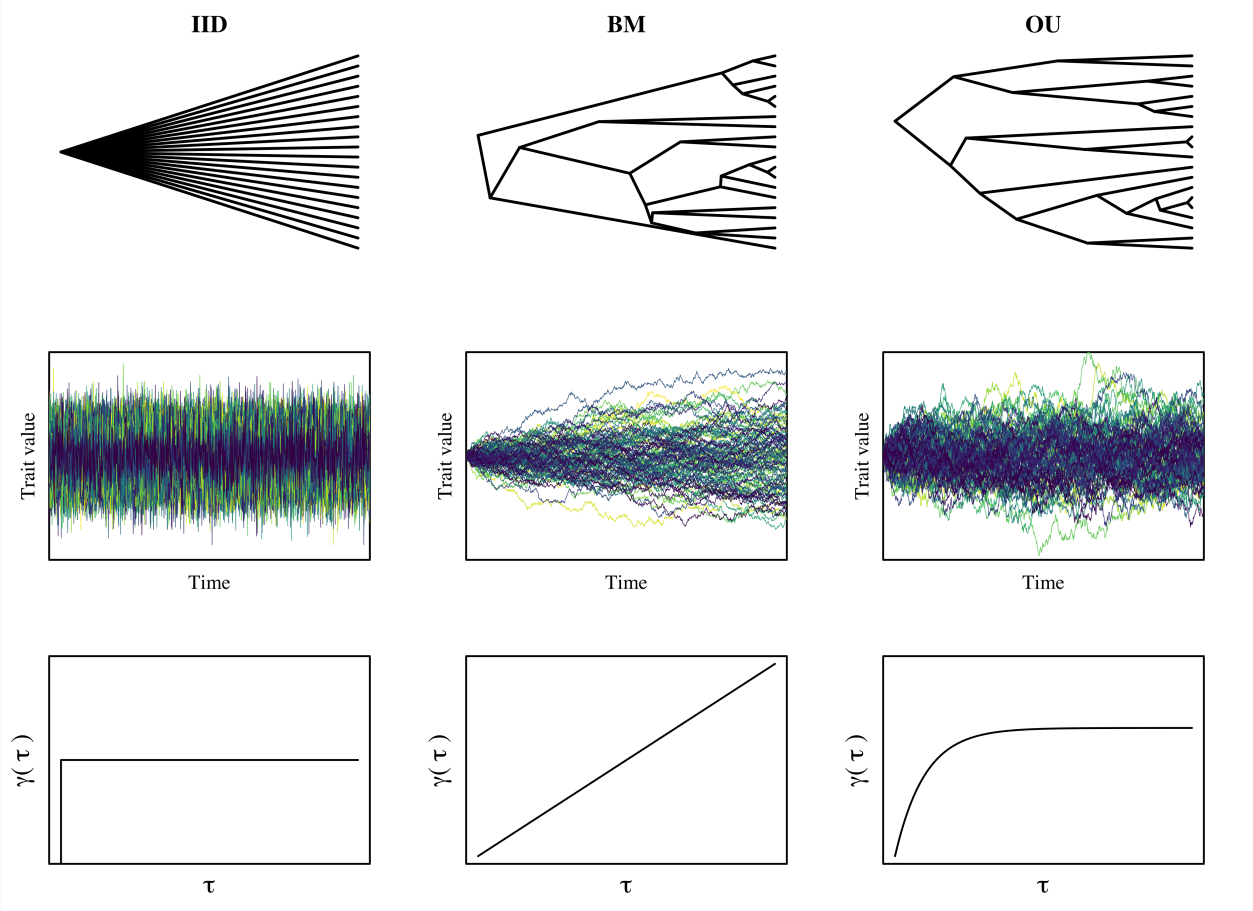
Phylogenetic correlation structures can be added via the `R` package `ape`.

Just like spatial and temporal autocorrelation, there are a number of alternatives to chose from:

- `corBrownian` Brownian motion model (Felsenstein 1985)
- `corPagel` The cov. matrix defined in Freckelton et al. (2002)
- `corMartins` The cov. matrix defined in Martins and Hansen (1997)
- `corGrafen` The cov. matrix defined in Grafen (1989)
- `corBlomberg` The cov. matrix defined in Blomberg et al. (2003)

Visualising Phylogenetic Autocorr.

R package `ctpm`



Source: Noonan et al. 2021

GLMs

GLM review

We saw how GLMs offer a powerful framework for modelling data types that can not be assumed to have Gaussian distributed errors.

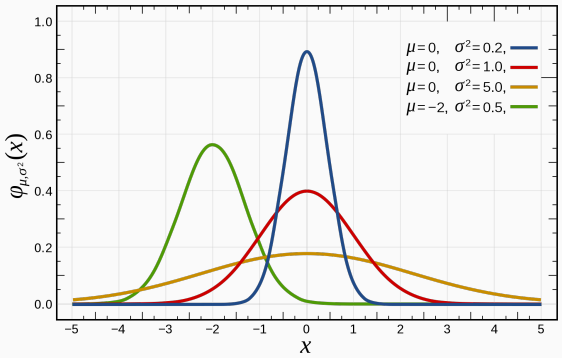
We then saw how, when fitting GLMs, we need to carry out 3 steps:

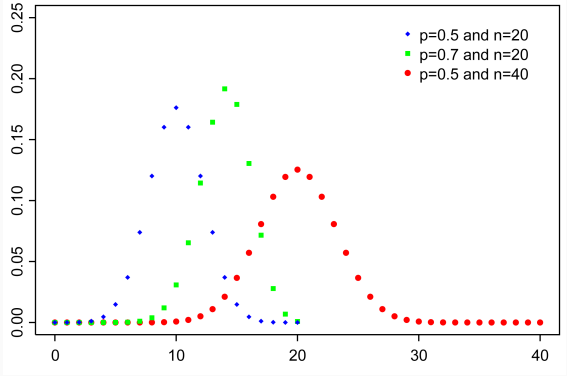
1. Make a distributional assumption on the response variable Y_i. This also defines the mean and variance of Y_i.
2. Specify the deterministic part of the model.
3. Formally specify the 'link' between the mean of Y_i and the deterministic part based on your distributional assumption.

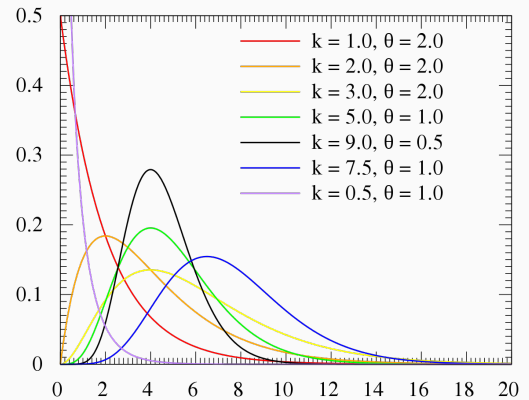
We then saw how to fit GLMs to count data in `R` using the `glm()` function.

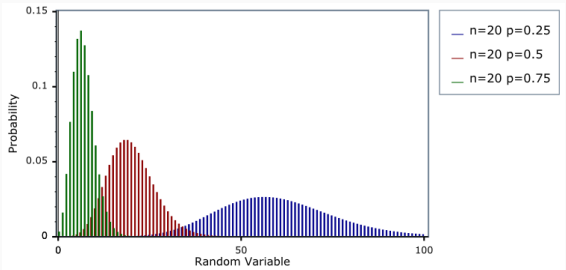
GLM review cont.

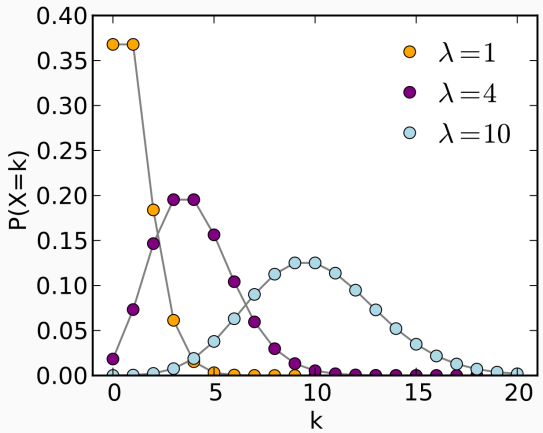
Because R functions streamline the process of fitting GLMs, the key step that's left in your hands is knowing when you will need to switch from a Gaussian model to a GLM, and identifying the correct distribution

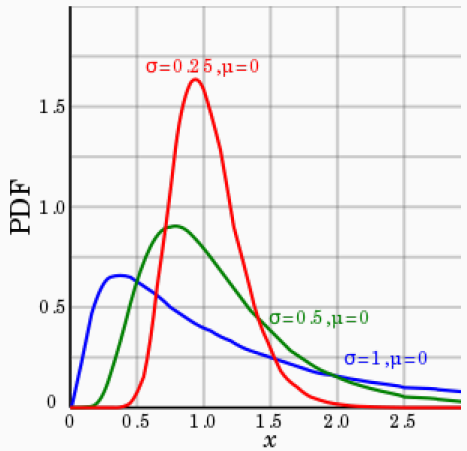












What we didn't cover

We covered standard GLMs, but this framework can be extended to handle nested data structures just like mixed effects models did for Gaussian linear regression. (a good resource for GLMMs: http://bbolker.github.io/mixedmodels-misc/glmmFAQ.html)

We covered log links and logit links, but there are a number of different link functions that you can use when fitting GLMs.

We also saw how modifying the off-diagonals of the correlation matrix can correct for various forms of autocorrelation in Gaussian linear regression, but because we're working with different distributions now those approaches don't translate cleanly.

Simulations and non-linearity

Simulations

Simulations can put practicing biologists on equal footing with experienced mathematicians/statisticians. This makes them potentially powerful tools for understanding biological systems and generalising the results of our analyses.

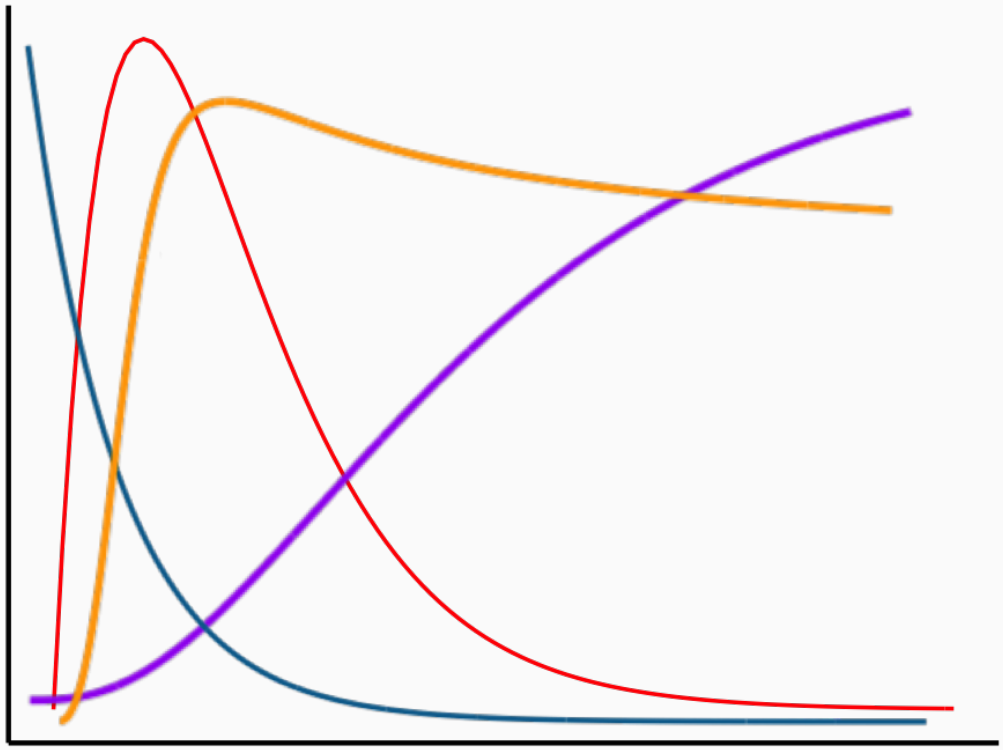
Simulations can also help us understand how a system might be expected to respond to conditions that we can not/couldn't measure.

Always remember that the more moving pieces a simulation setup has, the harder the outcomes can be to understand. Carefully tailored simulations are often more informative than complex simulations that we can't keep track of.

Simulations can also help us understand the statistical power of our data/model, but good experimental design is more important than simulation based power analysis (don't overthink it).

Biology is not always linear

Biological systems are not always linear, and you will need to become familiar with a wide range of deterministic functions.



Non-Linear Models

Non-Linear Models

The `nls()` function allows modelling non-linear relationships.

We covered some of the most common functions, but the full list of possibilities is infinite.

The better you get building a working knowledge of deterministic functions, the better you will get at building models to fit and make theoretical predictions (very useful knowledge to have in your tool-belt).

If you combine these functions with a stochastic model and maximum likelihood estimation you can fit any model you can write down to data.

References

Burnham, K.P., Anderson, D.R. & Huyvaert, K.P. (2011). AIC model selection and multimodel inference in behavioral ecology: some background, observations, and comparisons. *Behavioral Ecology and Sociobiology*, 65, 23–35.

Liang, M., Liu, X., Parker, I.M., Johnson, D., Zheng, Y., Luo, S., Gilbert, G.S. & Yu, S. (2019). Soil microbes drive phylogenetic diversity-productivity relationships in a subtropical forest. *Science advances*, 5, eaax5088.

Bolker, B. M. (2008). Ecological models and data in R. Princeton University Press.

Zuur, Alain F., Elena N. Ieno, Neil J. Walker, Anatoly A. Saveliev, and Graham M. Smith. 2009. Mixed Effects Models and Extensions in Ecology with R, 261–93. New York, NY: Springer New York.