

Experimental design

- Asking good questions
- Samples and populations
- Confounding
- Controls
- Replication
- Independence

What is the answer to the ultimate question of life, the universe, and everything?

Asking good questions

All experiments are designed to answer a question. The question (or hypothesis) defines everything about the experiment; duration, spatial extent, measurement units, replication etc.

It is very important to ask questions which are specific and answerable. Some things to think about are the study **population**, and how it is related to natural or experimental **predictors**.

Study population

- What is the **scope** of the study, in time, space and ecologically?
- What is the **response** (dependent variable)?

Predictors

- What are the **predictors** (independent variables)?
- What is the hypothesised **relationship** between the predictors and response?

Be clear about your study population.

Question

Are there any differences between animal abundance at different altitudes?

What is the response (dependent variable)?

What is the scope in time, space and ecologically?

Better question

Are there differences between the number common wombats (*Vombatus ursinus*) caught in traps at Kosciusko National Park in the summer between 2017 and 2012 at different altitudes?

Be clear about predictors and their relationships with the response.

Question

Are there differences between the number common wombats (*Vombatus ursinus*) caught in traps at Kosciusko National Park in the summer between 2017 and 2012 at different altitudes?

What is the range of altitudes you are interested in?

What is the hypothesised relationship between altitude and number of wombats?

Better question

Does the number common wombats (*Vombatus ursinus*) caught in traps decrease with altitude (between 1000 and 2000m above sea level) in Kosciuszko National Park in summer between 2017 and 2020?

Samples and populations

When conducting experiments, we collect samples, and try to make inferences about the population.

We select 5 geese randomly from a lake and find 4 of them are male.

What is the population?

What can we conclude?

Are there more males than females in the population?

Biologists generally want to make inferences (draw conclusions) about a (statistical) population, rather than a sample.

Biological population - a group of individuals from the same species

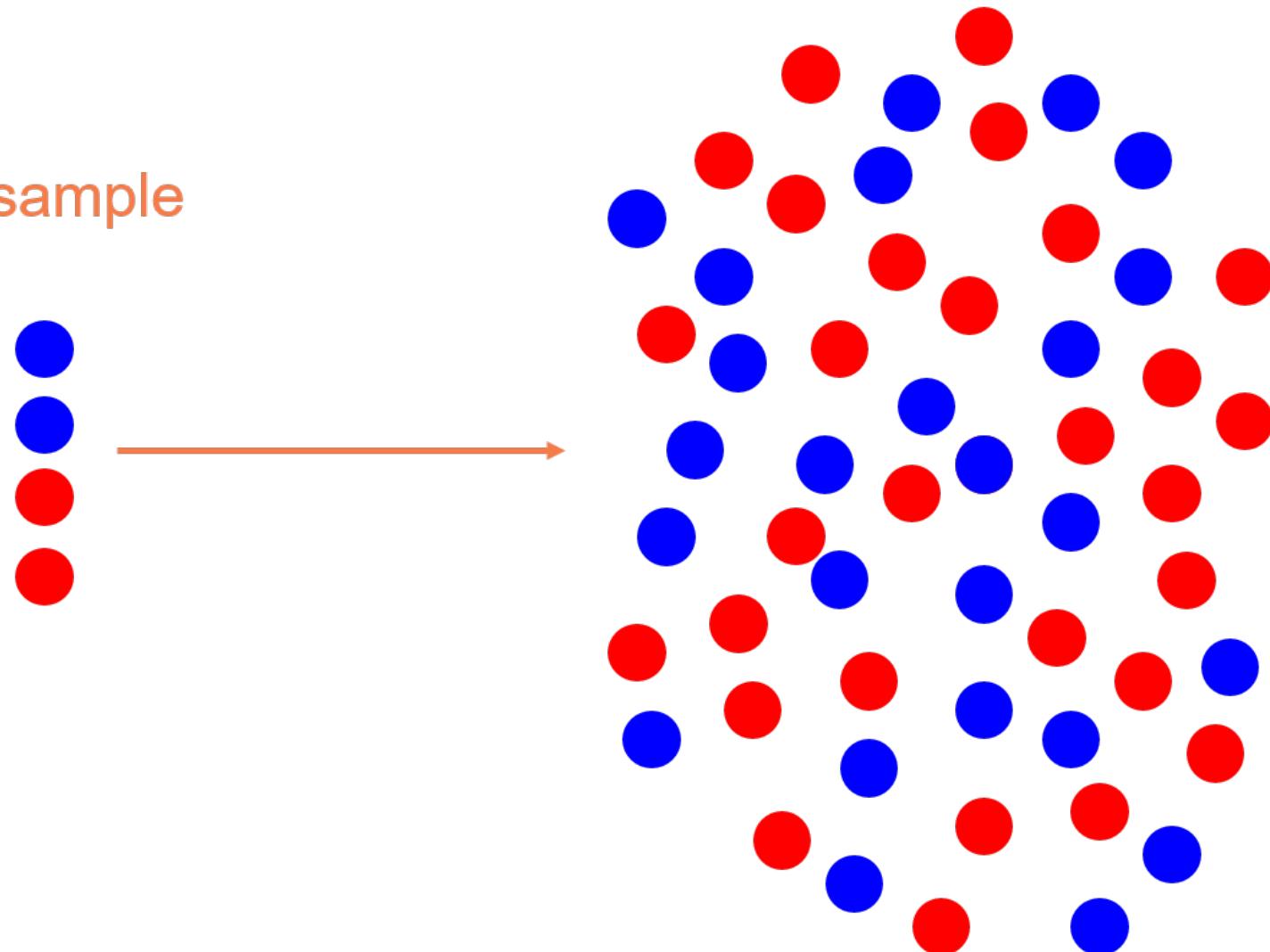
Statistical population - the collection of all possible observations, this can be the same or distinct from a biological population. Characteristics of populations are called parameters (e.g. population mean)

Sample - The collection of observations. Characteristics of samples are called statistics (e.g. sample mean)

Any statistic of a sample will differ from the true population parameter.

likely population

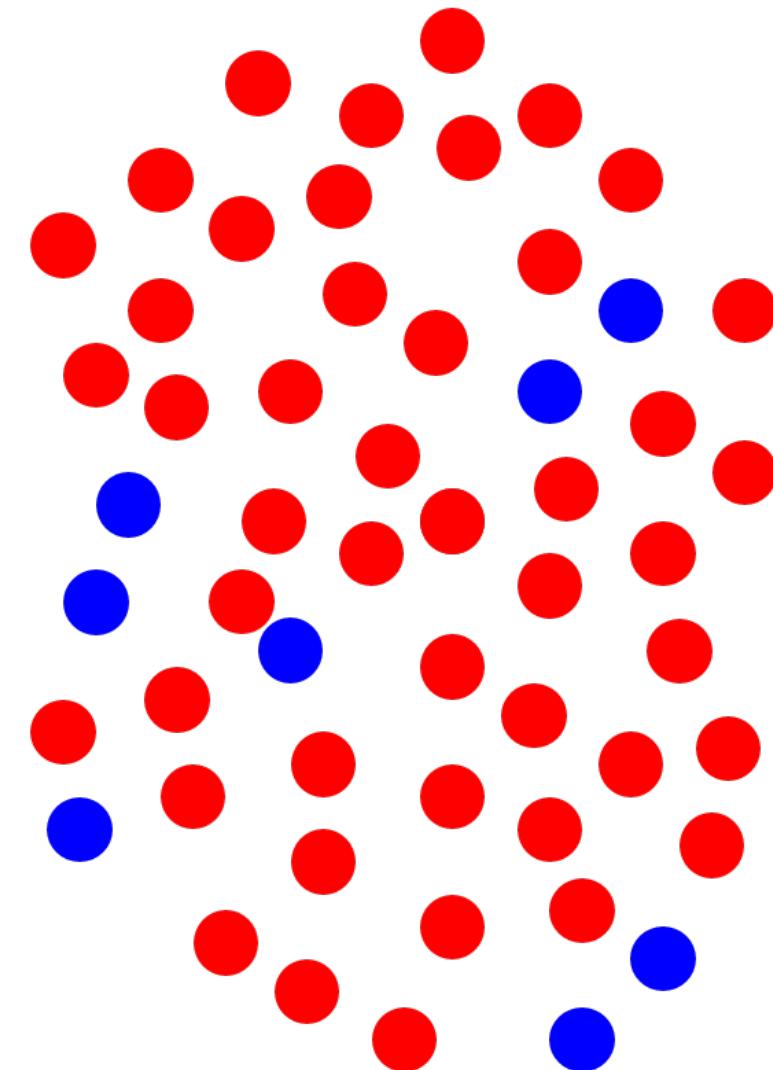
known sample



known sample

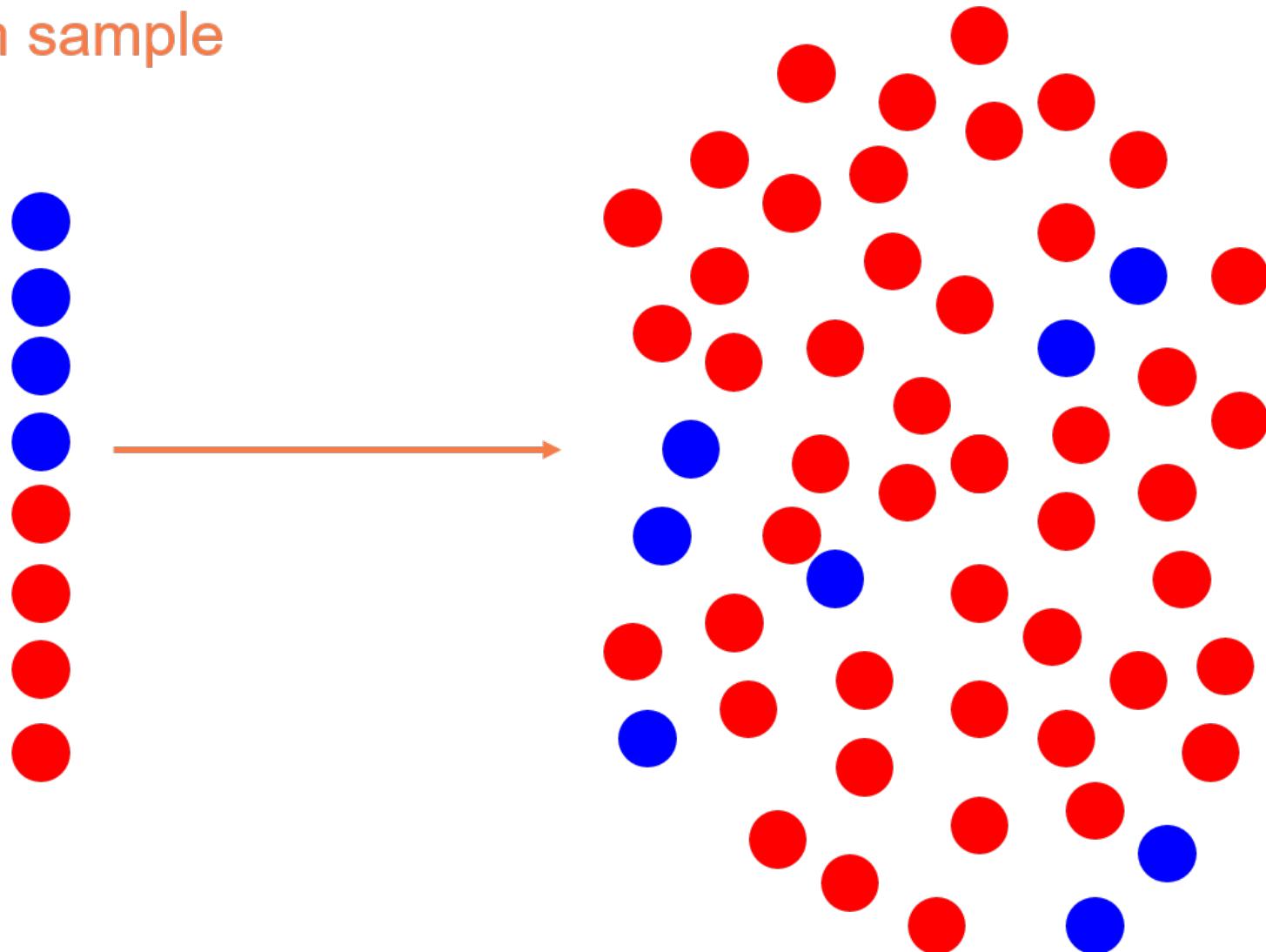


possible population



less likely population

known sample



Confounding

Confounding means that differences due to experimental treatments cannot be separated from other factors that might be causing the observed differences.

Example

If you measure the height of children and their maths ability, you would conclude that taller children are better at maths.

What is the confounding variable?

Confounding is often quite subtle, and should always be thought about when planning an experiment. Experiments with good replication and controls can remove all sources of confounding, we will look into these in detail in a moment.

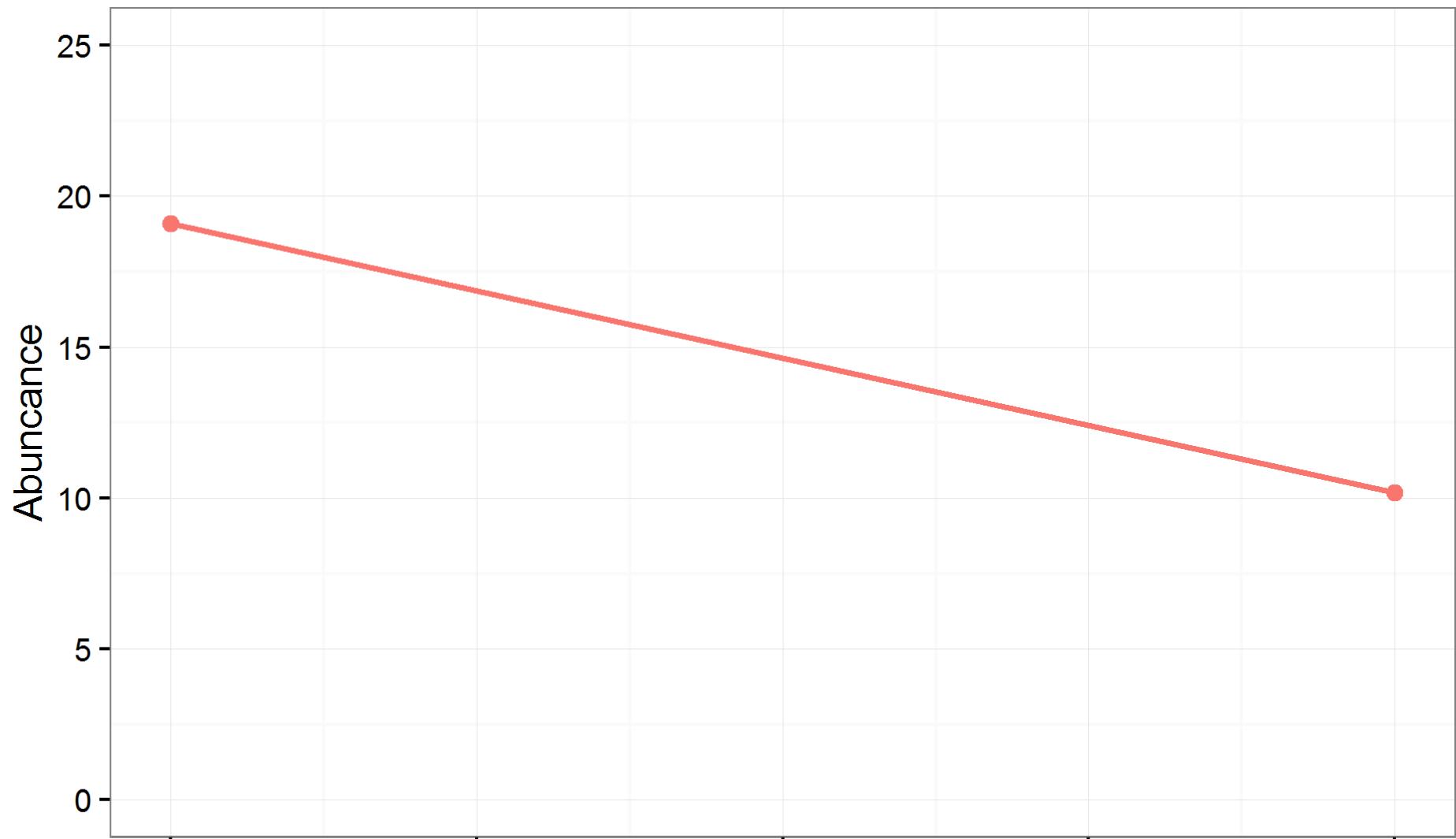
In addition we can include possible confounding variables in the analysis, also known as "controlling for" these variables. As it is never possible to know all the variables that could be confounding your effects, this strategy is less desirable than having good controls and replication.

Controls

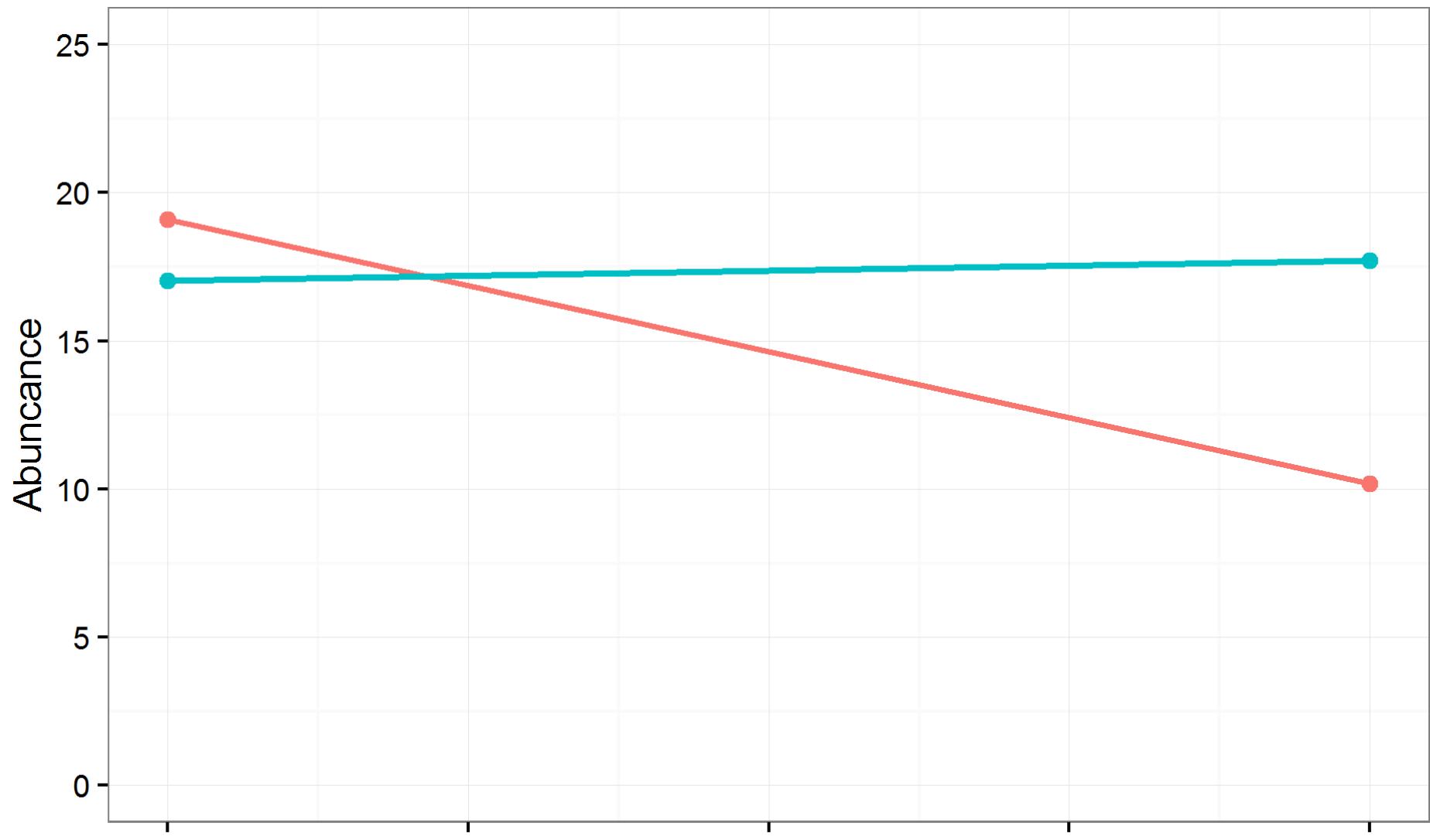
Example

An ecologists measures the abundance of Murray cod downriver of a new golf course before it is built and again after it is operational, a year later. The abundance of cod halves on average over this time.

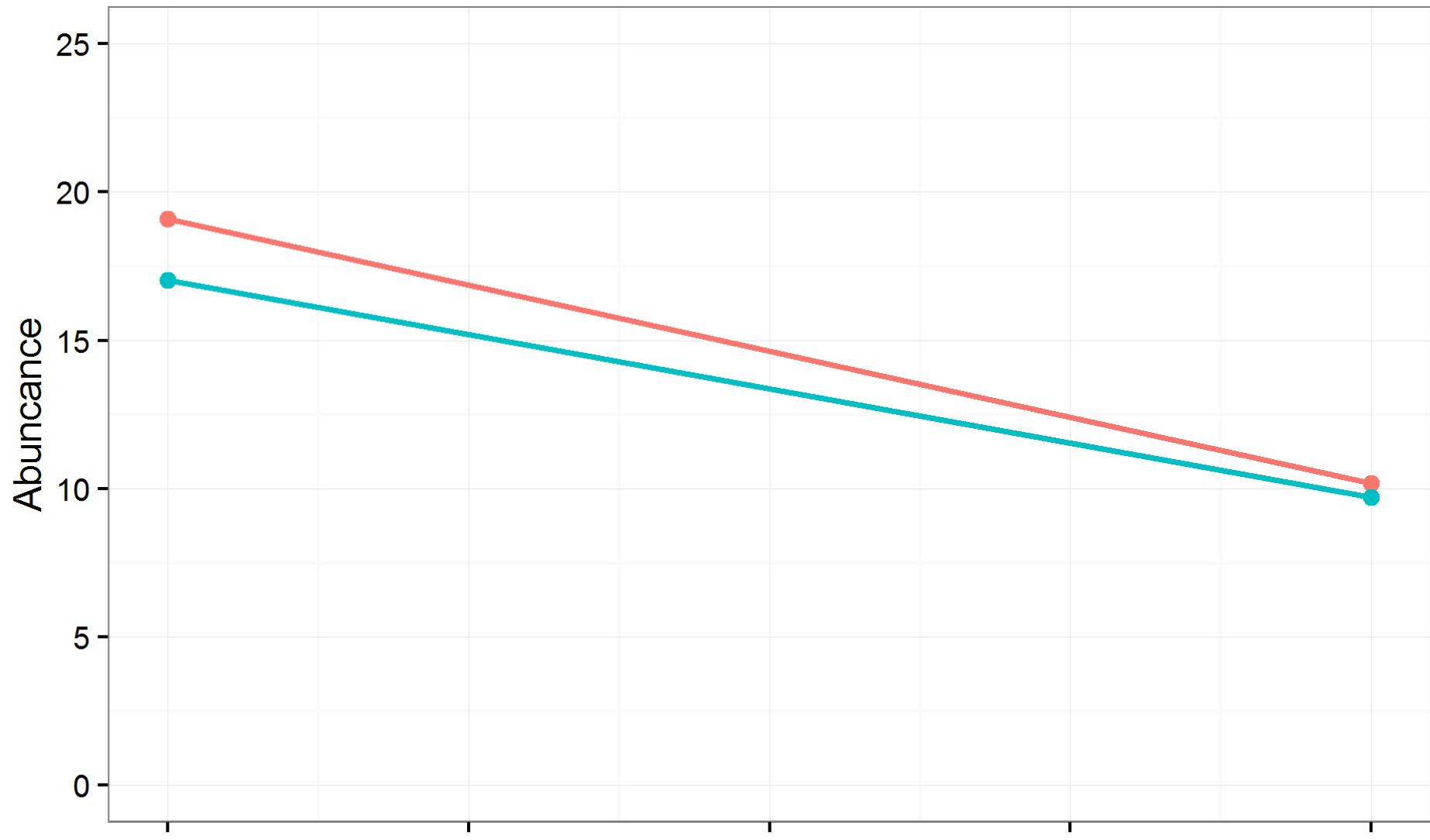
What can we conclude?



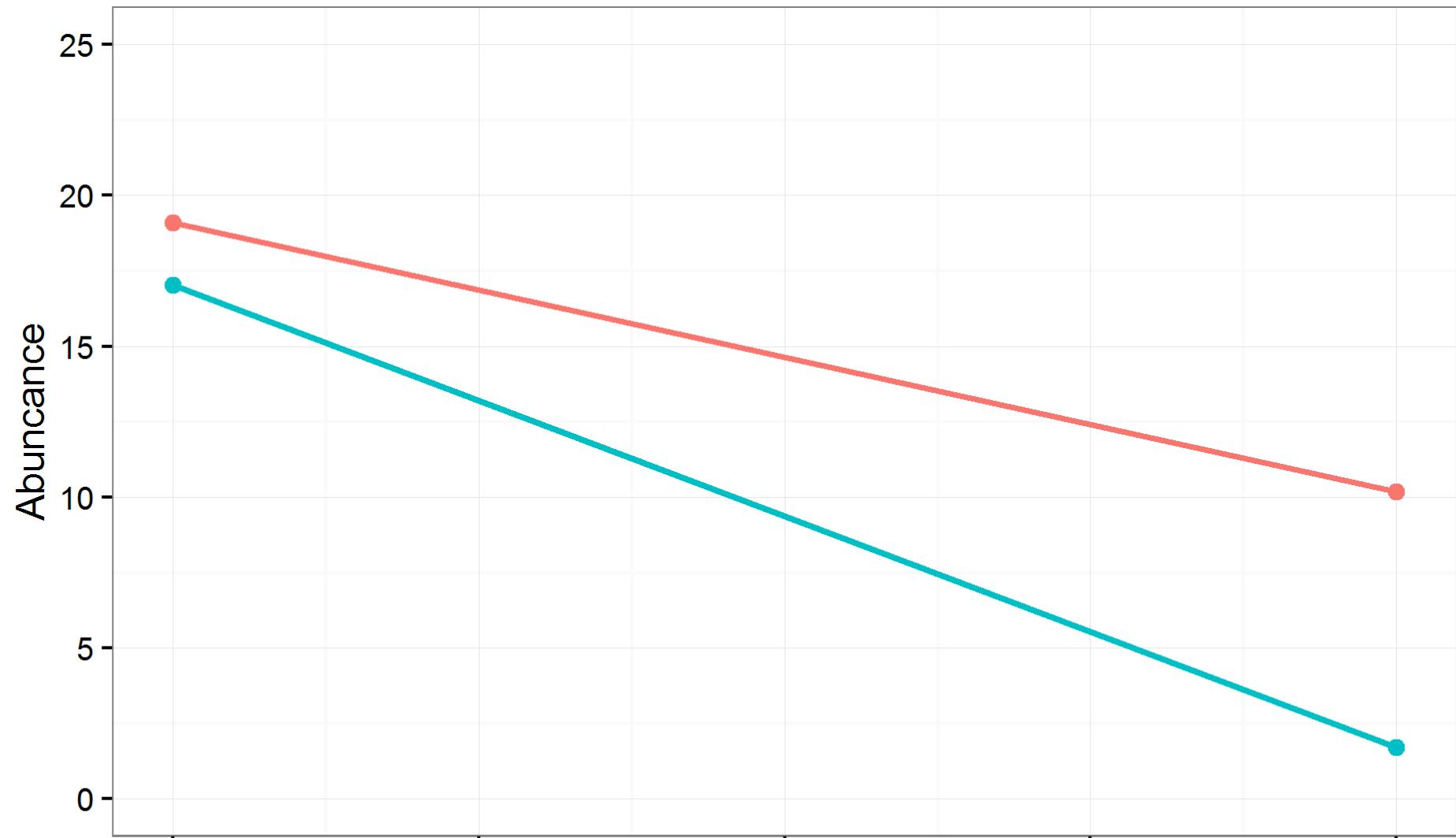
1.14



1.15



1.16



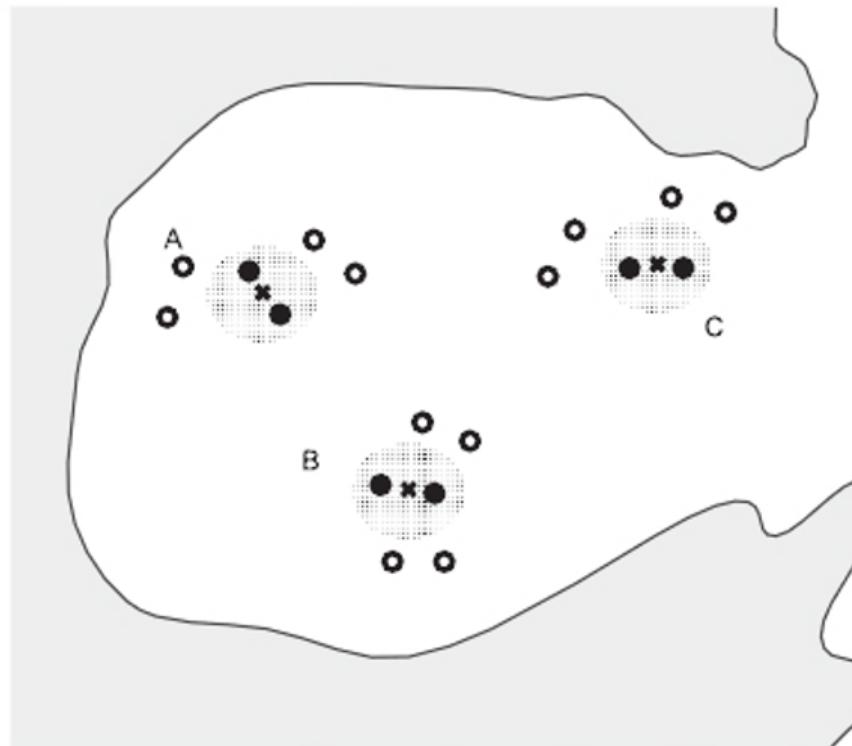
1.17

Many factors can influence the outcome of an experiment, like seasons, weather, politics, aliens. These things are often not under our control. It is essential that we know what would have happened if not for the experimental manipulation (treatment, impact) so we can compare what happened with the background changes in the population of interest.

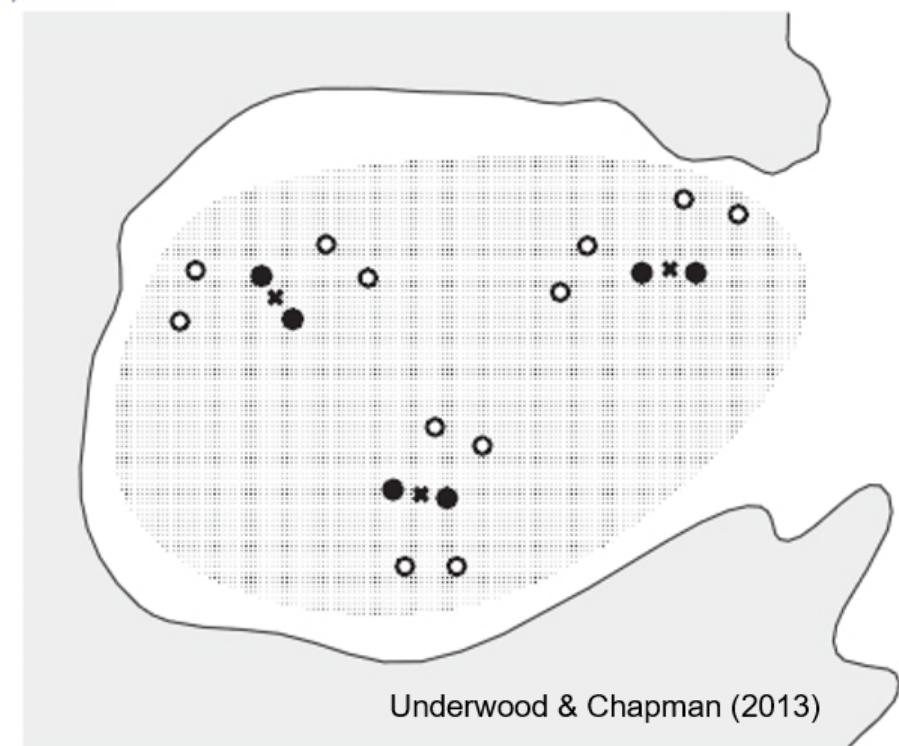
In the golf course example we had confounding between impact and time (and all the variables that change with time).

Deciding to have controls is not enough, the controls have to be well designed. Control and treatment units should be as similar as possible, except from the treatment. They also need to be at an appropriate scale and not be confounded.

(a) Predicted



(b) Actual



Underwood & Chapman (2013)

More on controls

We want to test the effect of predation on marine snails. To do this we exclude predatory fish using cages and compare the changes in caged and uncaged areas. We find a large difference in snail abundance between the caged and uncaged areas.

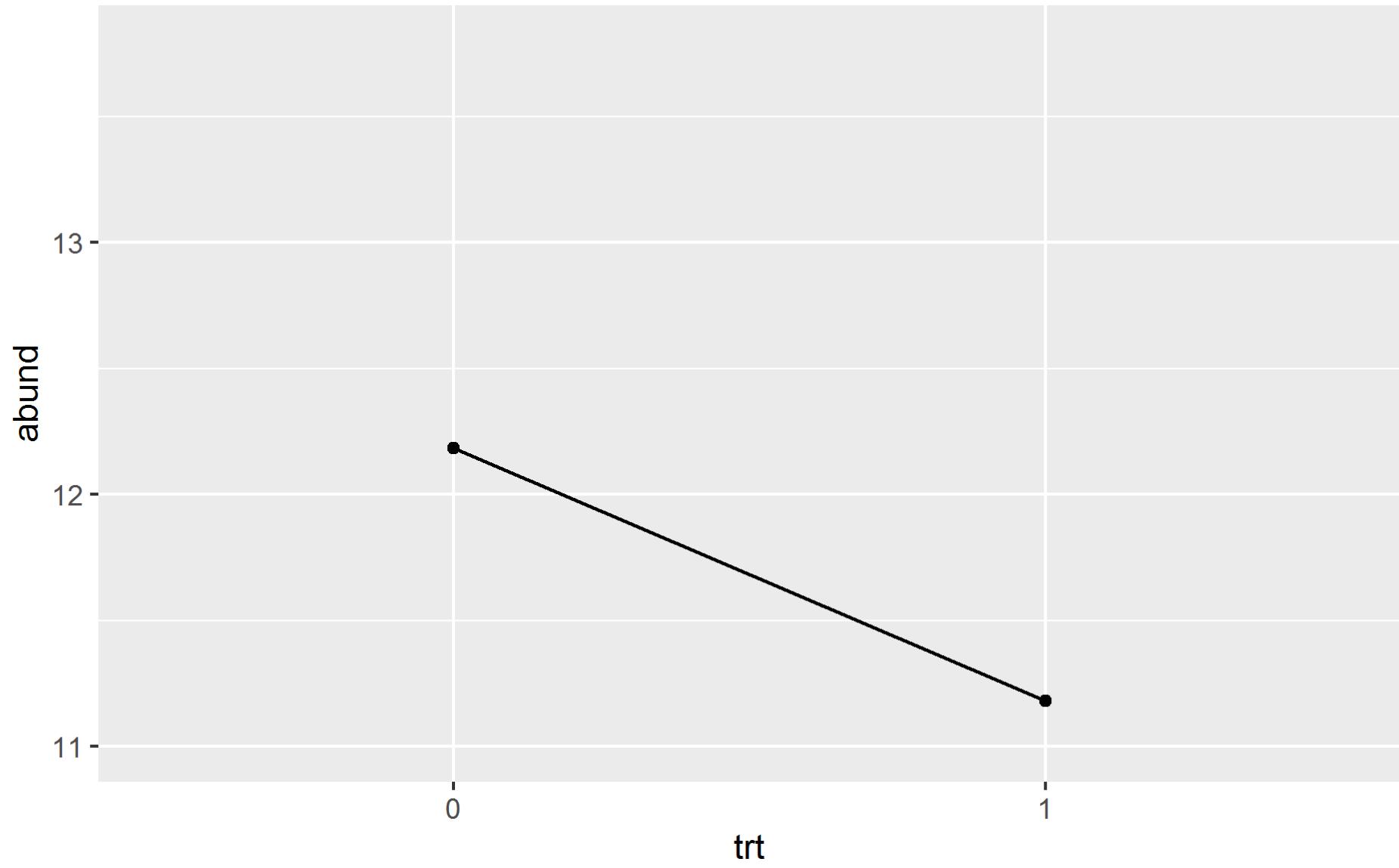
What can we conclude?

There may be an effect of predation, but the effect we observe could be due to a number of other reasons associated with the cage like shading or reduced water movement. This is a form of confounding.

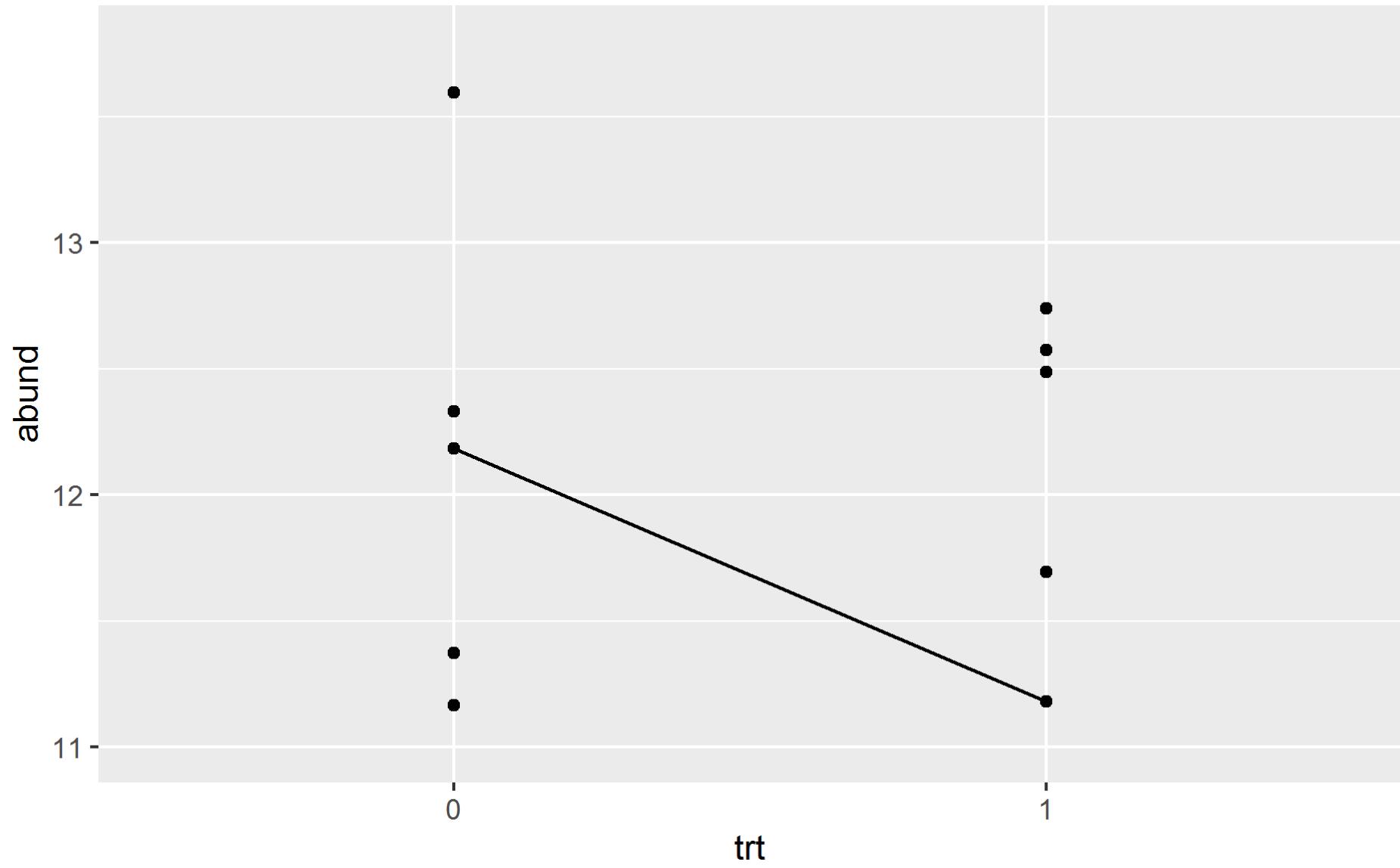
To be able to better attribute changes to predation, we can include procedural controls. In this case we may build an open cage, which is as similar as possible to the treatment cages, but allows predation, and use this as the control rather than completely uncaged areas.

Replication

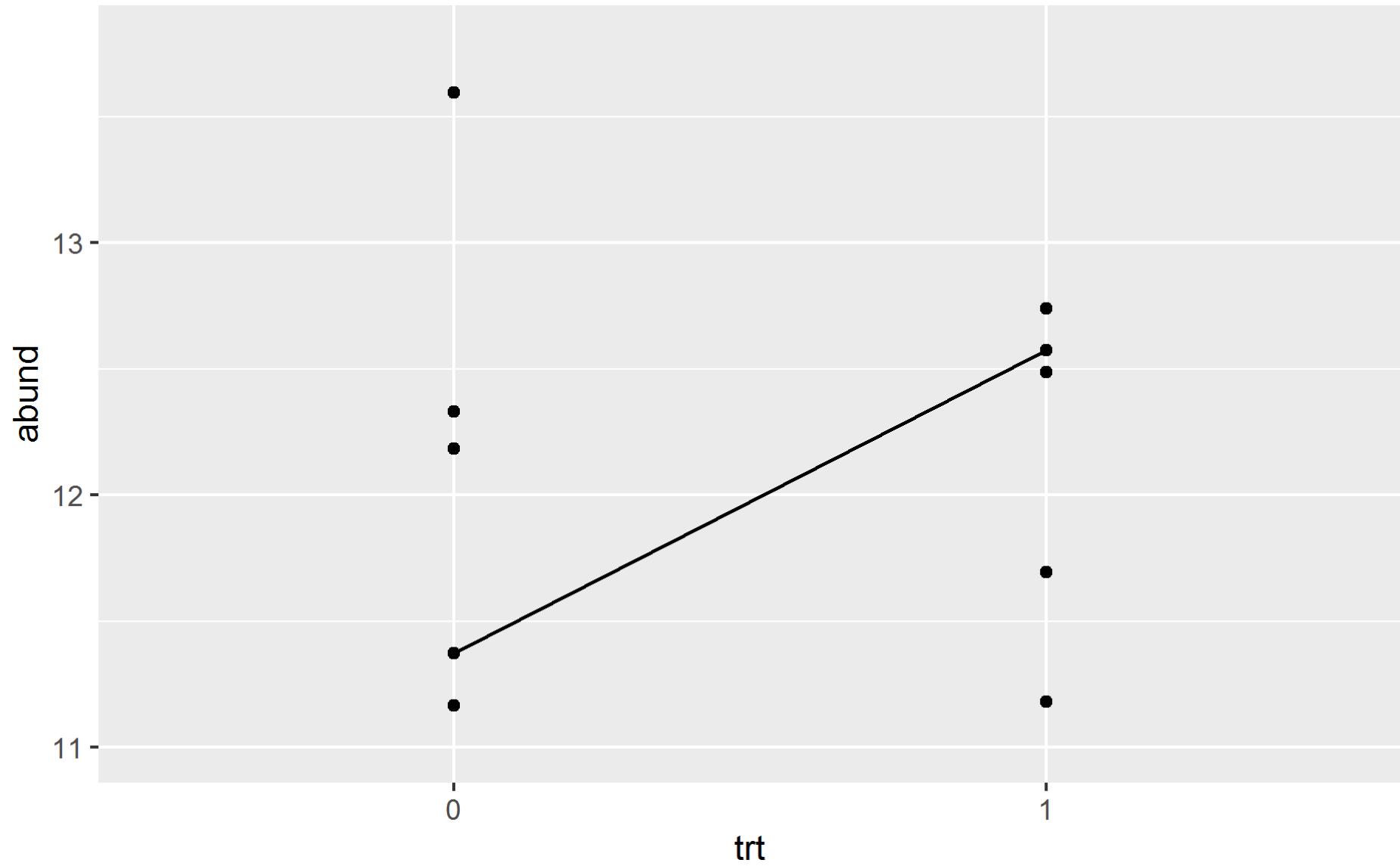
We cannot say measurements from two groups are different unless we know how much measurements in the same group vary. Some replication at the correct scale is needed for valid inference.



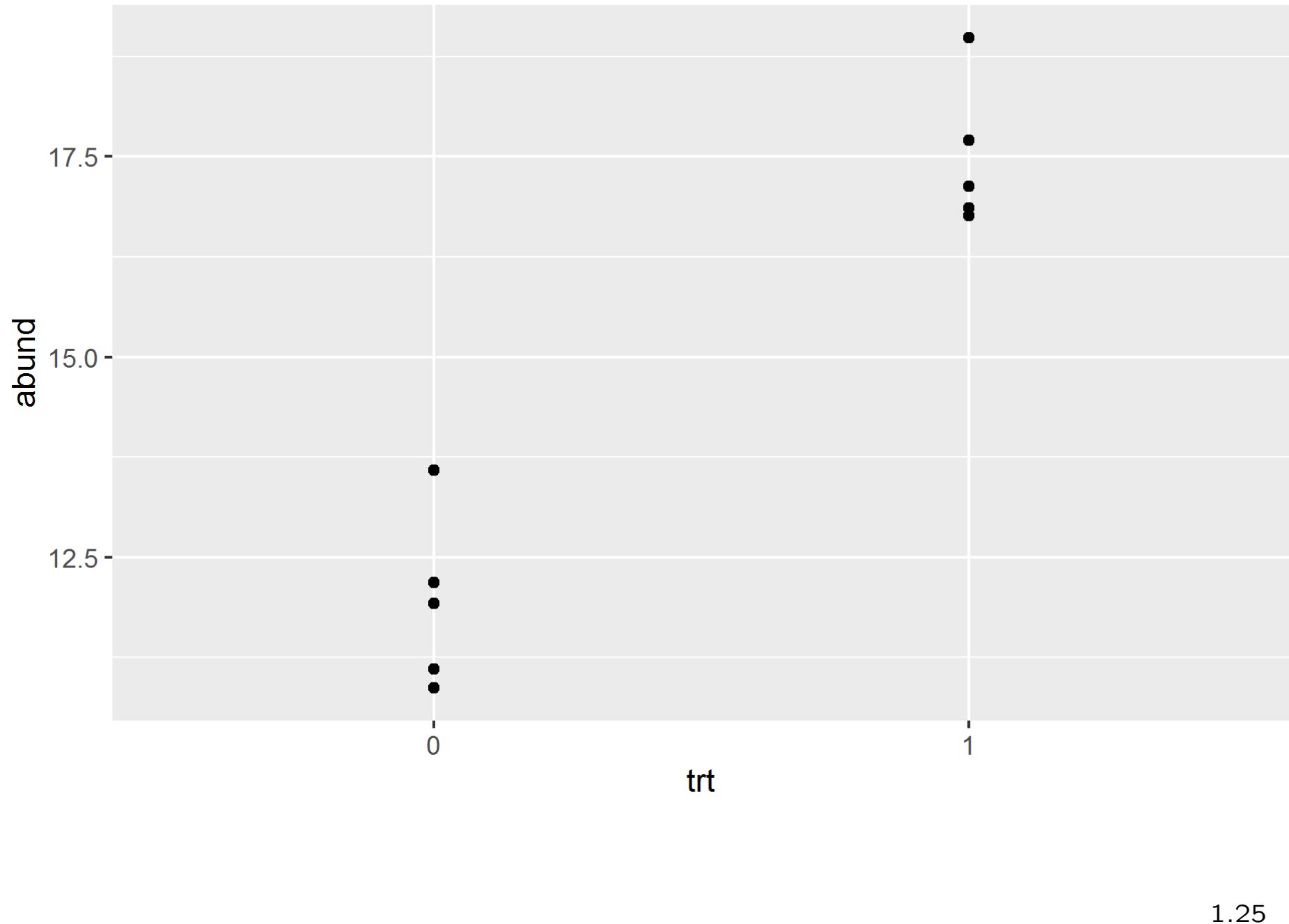
1.22

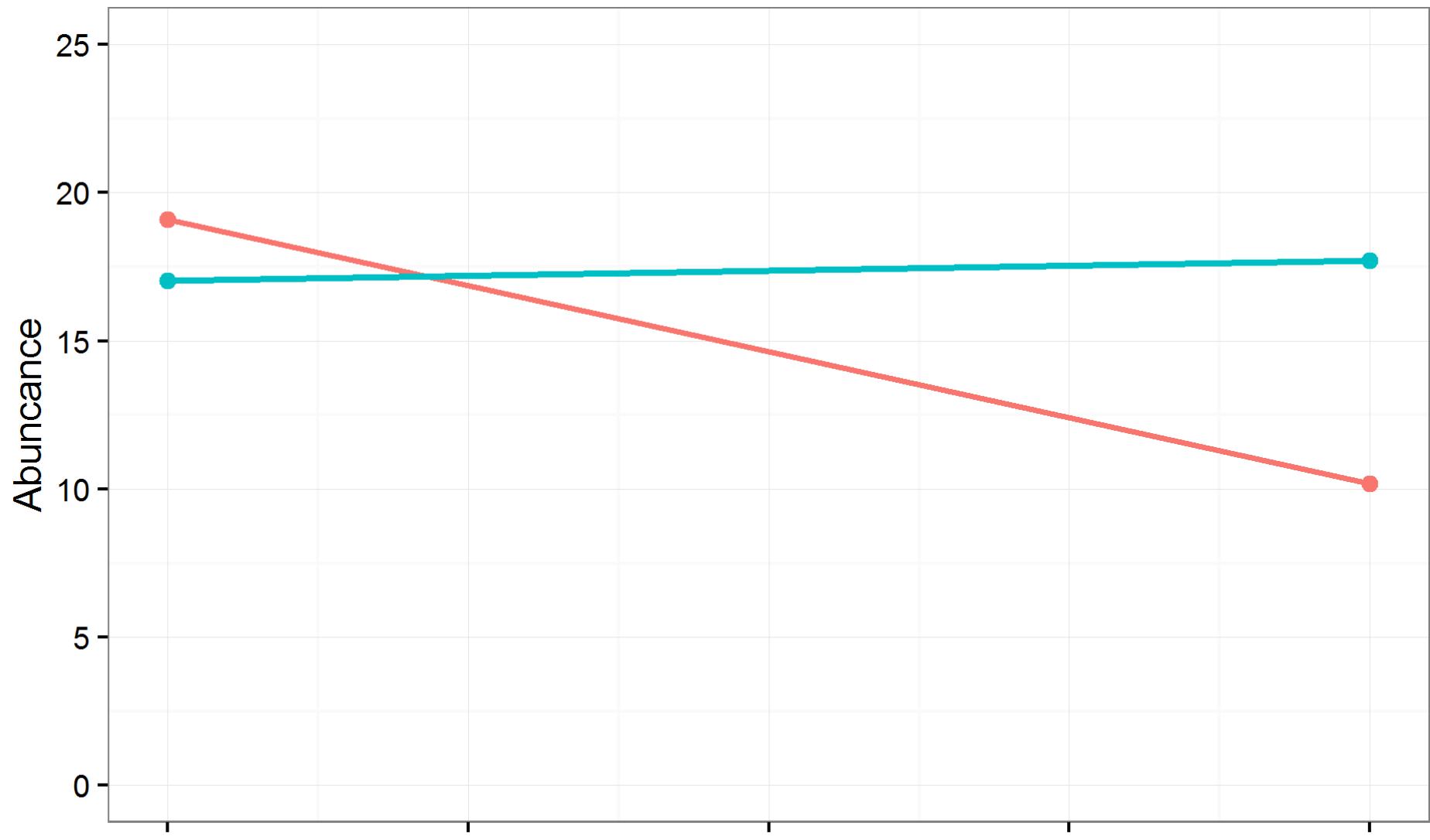


1.23

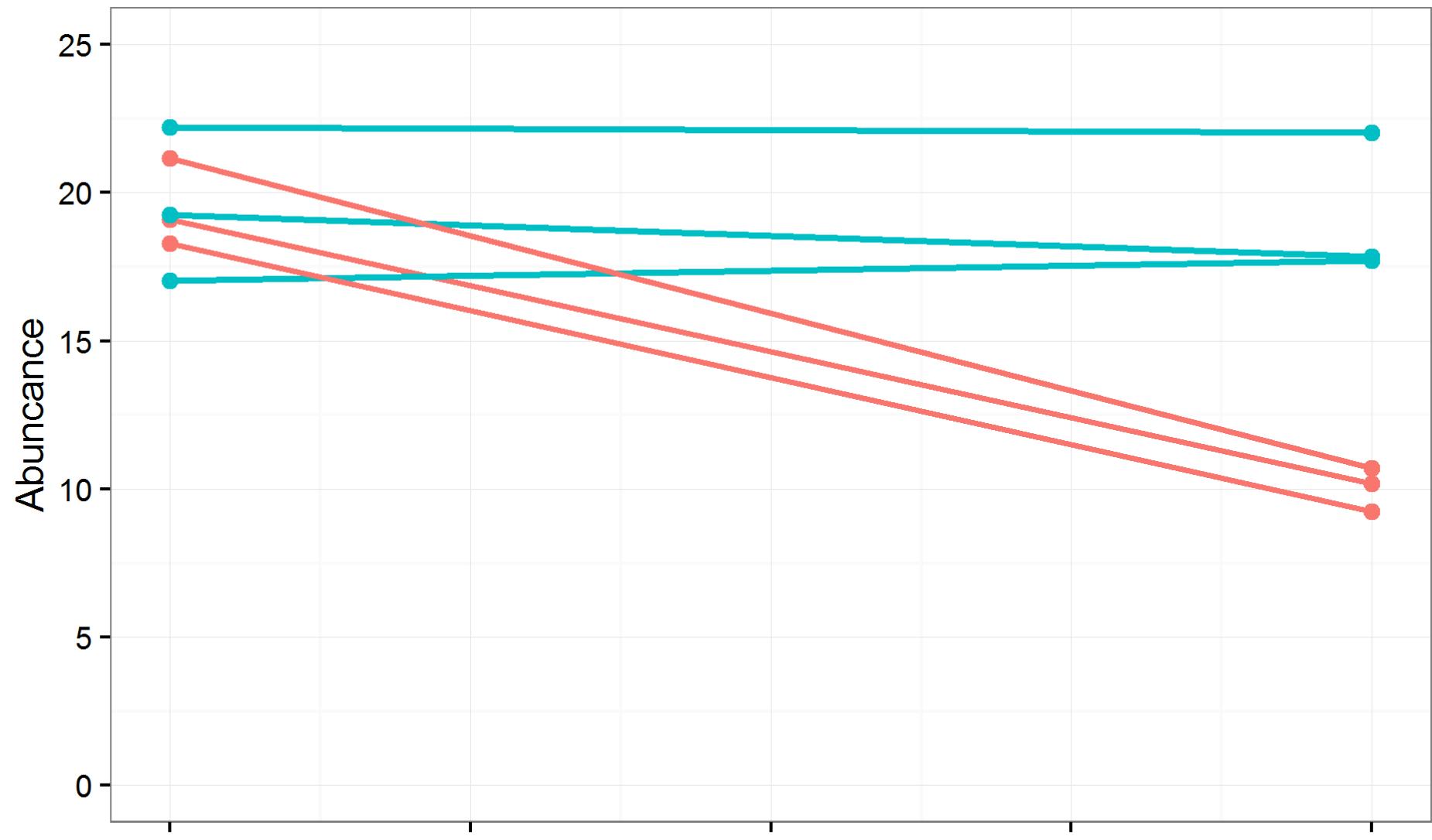


1.24

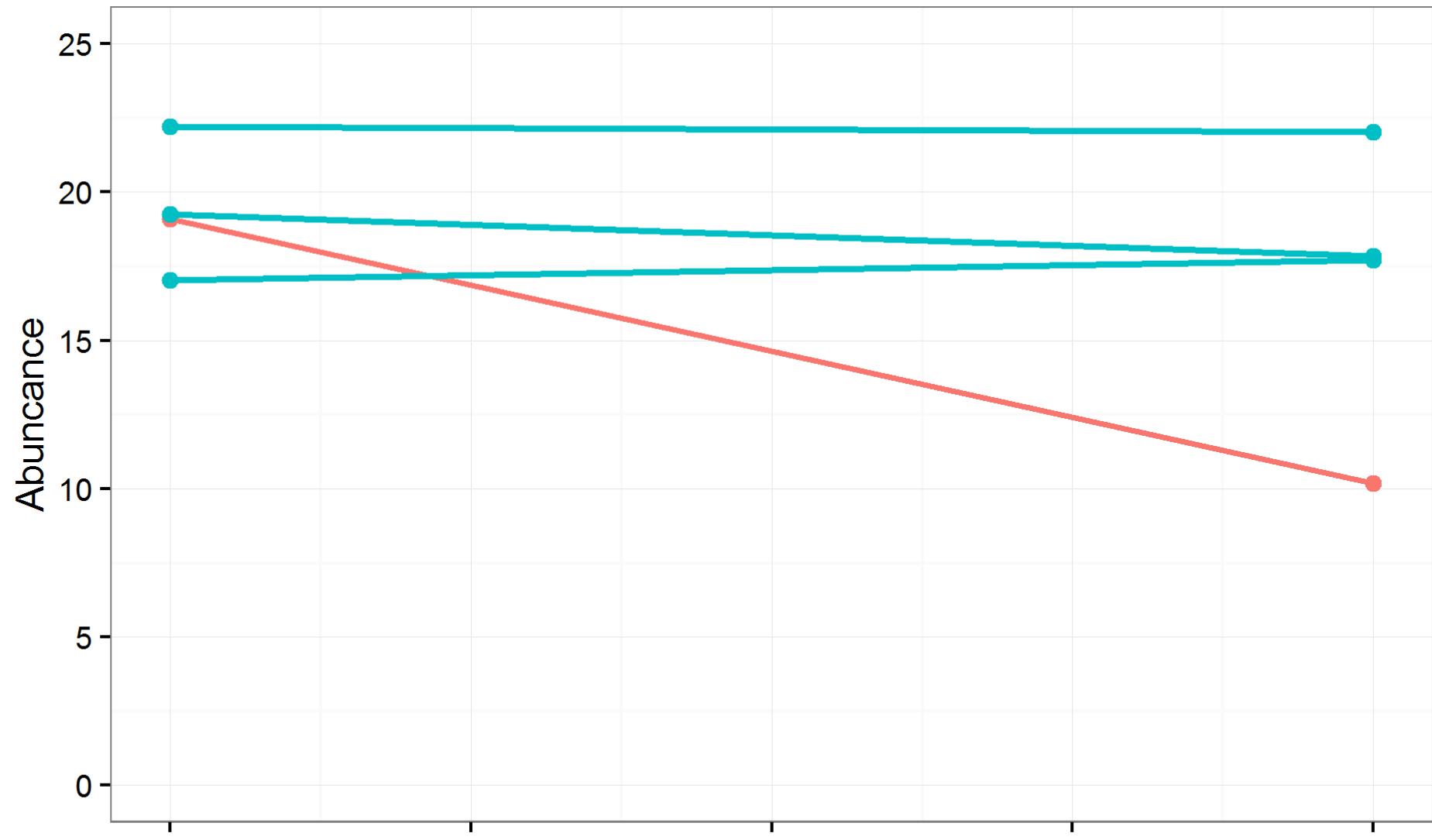




1.26



1.27



1.28

Independence

Independence between experimental units means knowing the value of one unit doesn't tell you anything about any other unit. Independence breaks down when some units are more similar than others.

Animals in the same aquarium are more similar (exposed to similar conditions) than those in different aquariums.

Measurements from the same animal are more similar than measurements from different animals.

Measurements taken from areas close to one another (in space or time) are more similar than measurements taken further apart.

This lack of independence (or dependence) must be taken into account in both experimental design and analysis. Sometimes having dependent samples is a deliberate strategy to estimate variation at different scales, more on this later.

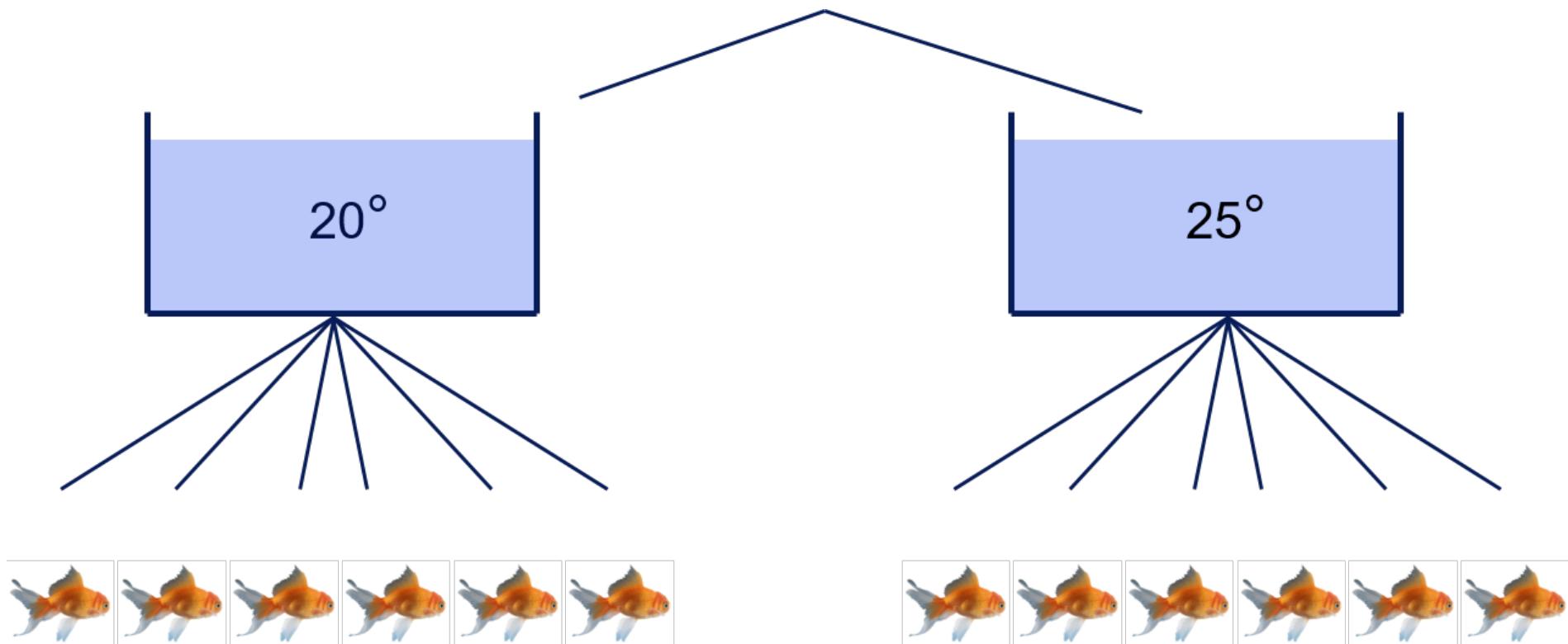
Pseudoreplication

Pseudoreplication looks and feels like replication, but experimental units are dependent. It results in invalid inference, and is functionally the same as no replication.

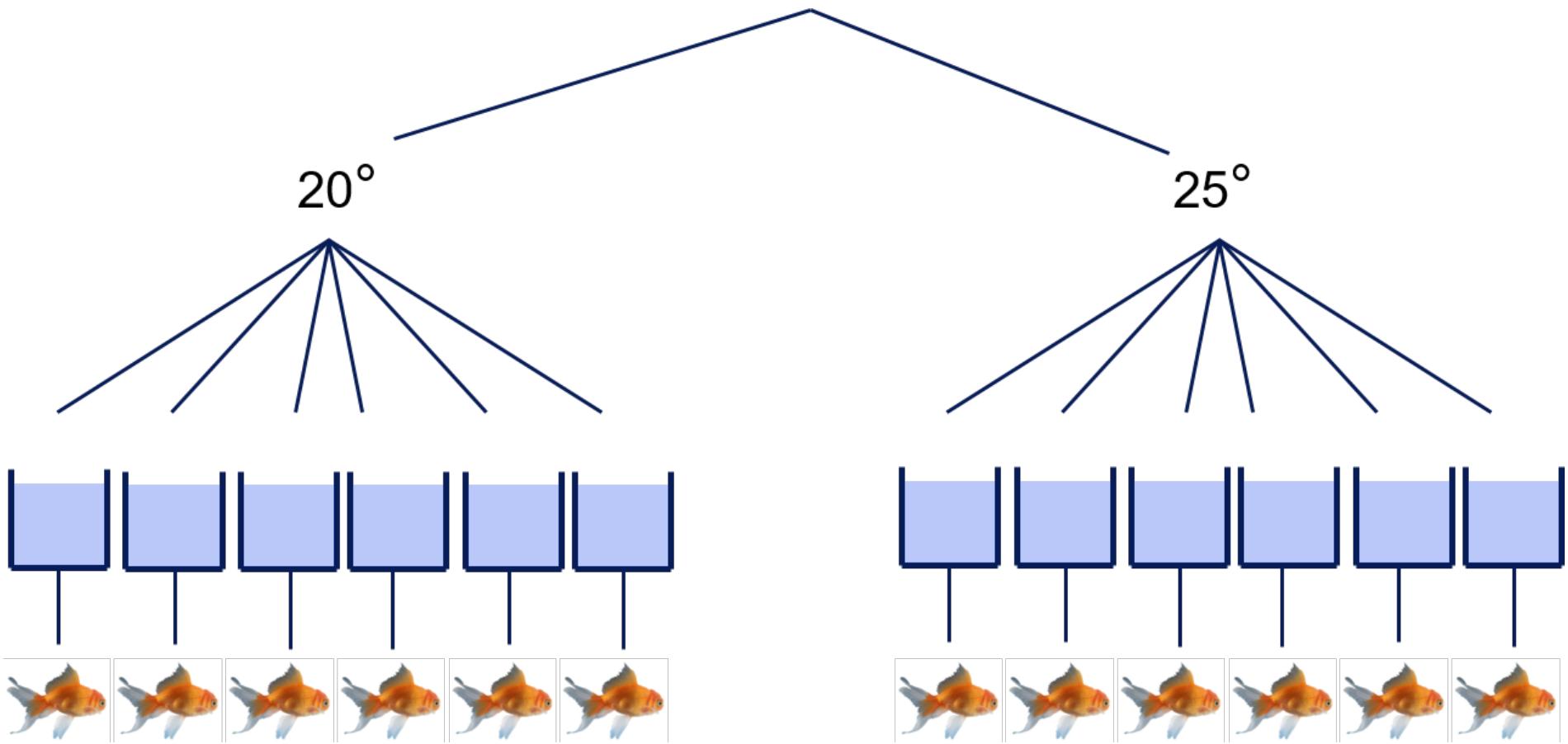
Example

We are interested in how temperature affects the speed of fish. We place 6 fish in a tank with a low temperature, and 6 fish in a tank with high temperature. We observe the fish in higher temperature swim faster on average.

What can we conclude?



Measurements are replicated, but not treatments.
Observed effect could be due to the tanks being different.



Independent replicates for each treatment.
Tanks randomly arranged in lab.

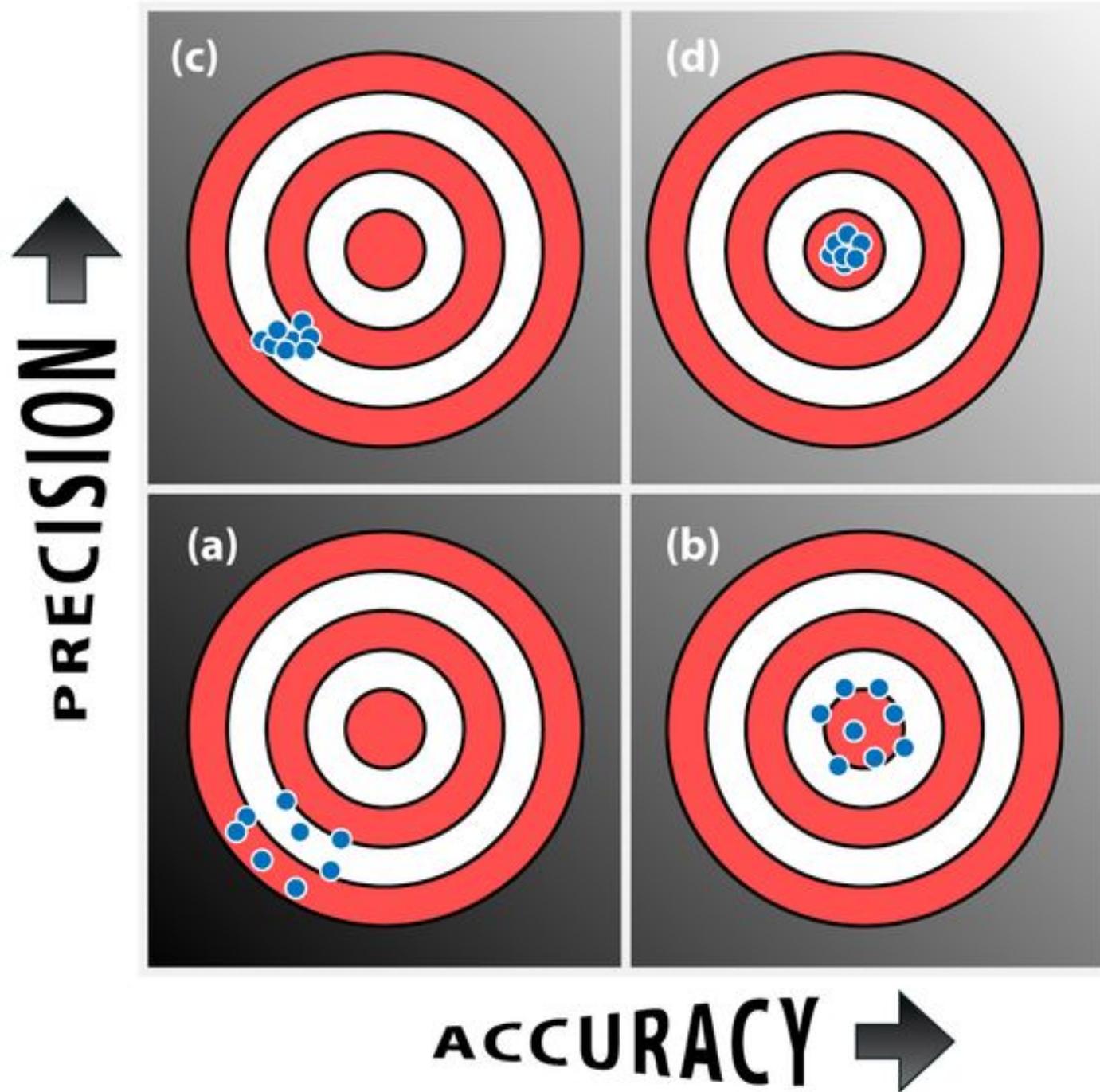
Sampling designs

- Sampling units
- Randomisation
- Simple random sampling
- Stratified sampling
- Blocking
- Random and fixed blocks
- Pilot studies

Good sampling designs ensure inference for population parameters using your sample is

Accurate rather than bias - close to the true value of the population

Precise rather than variable - close to one another



Sampling units

Sampling units are simply the quantities on which you take measurements. Often these are quite natural, for example in the geese example, each goose is a natural sampling unit.

At other times appropriate sampling units are not obvious, and choices can be made. In ecology we often use transects, quadrats, samples of soil, time and space as sampling units, and we have to make choices about the size and duration of each. It is important that these are well defined prior to sampling and in all reporting.

Any inference made is dependent on the sampling unit.

Example

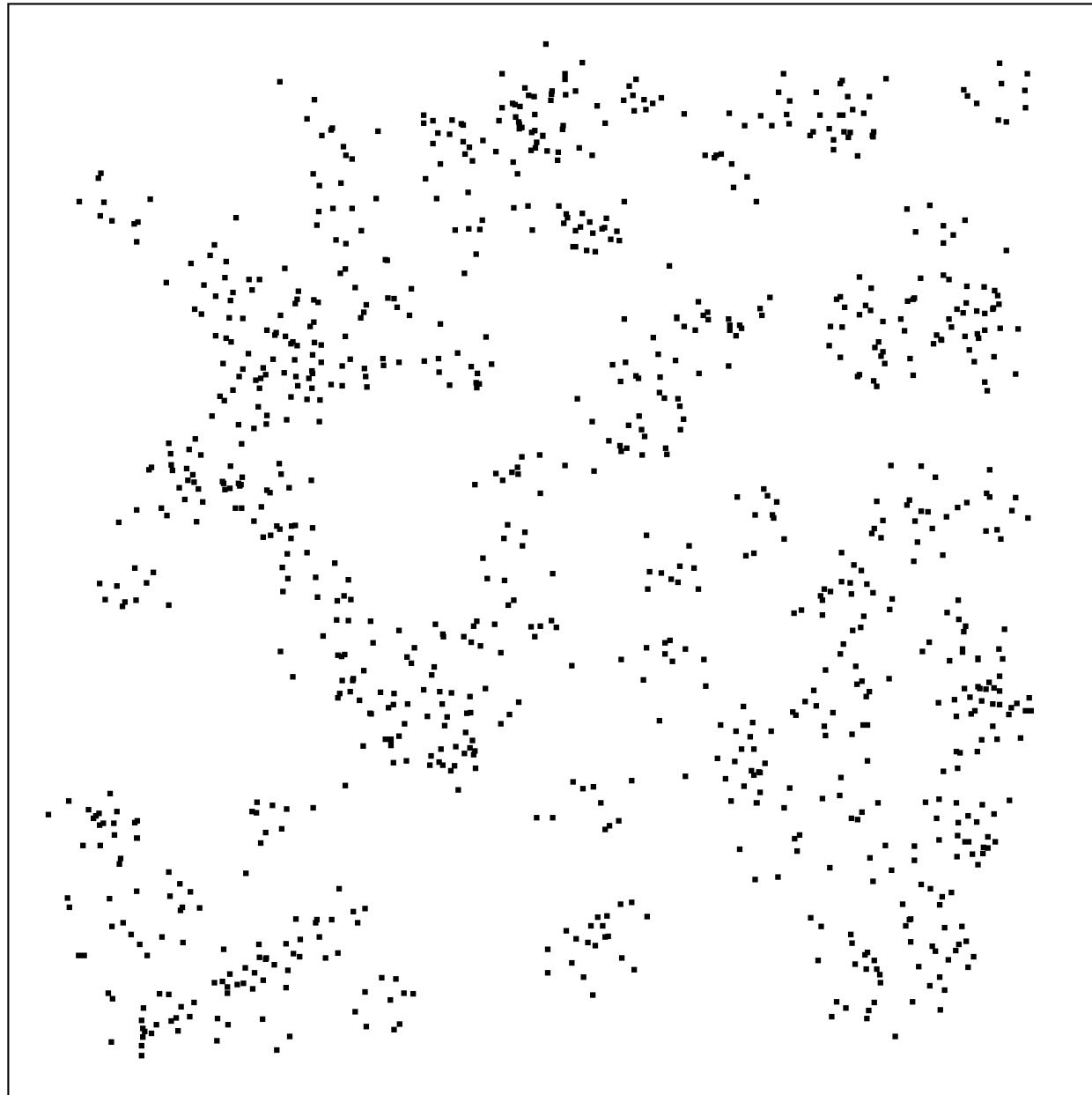
Sampling unit - A 10 cm x 10 cm plate submerged in 1 m of water for 1 month

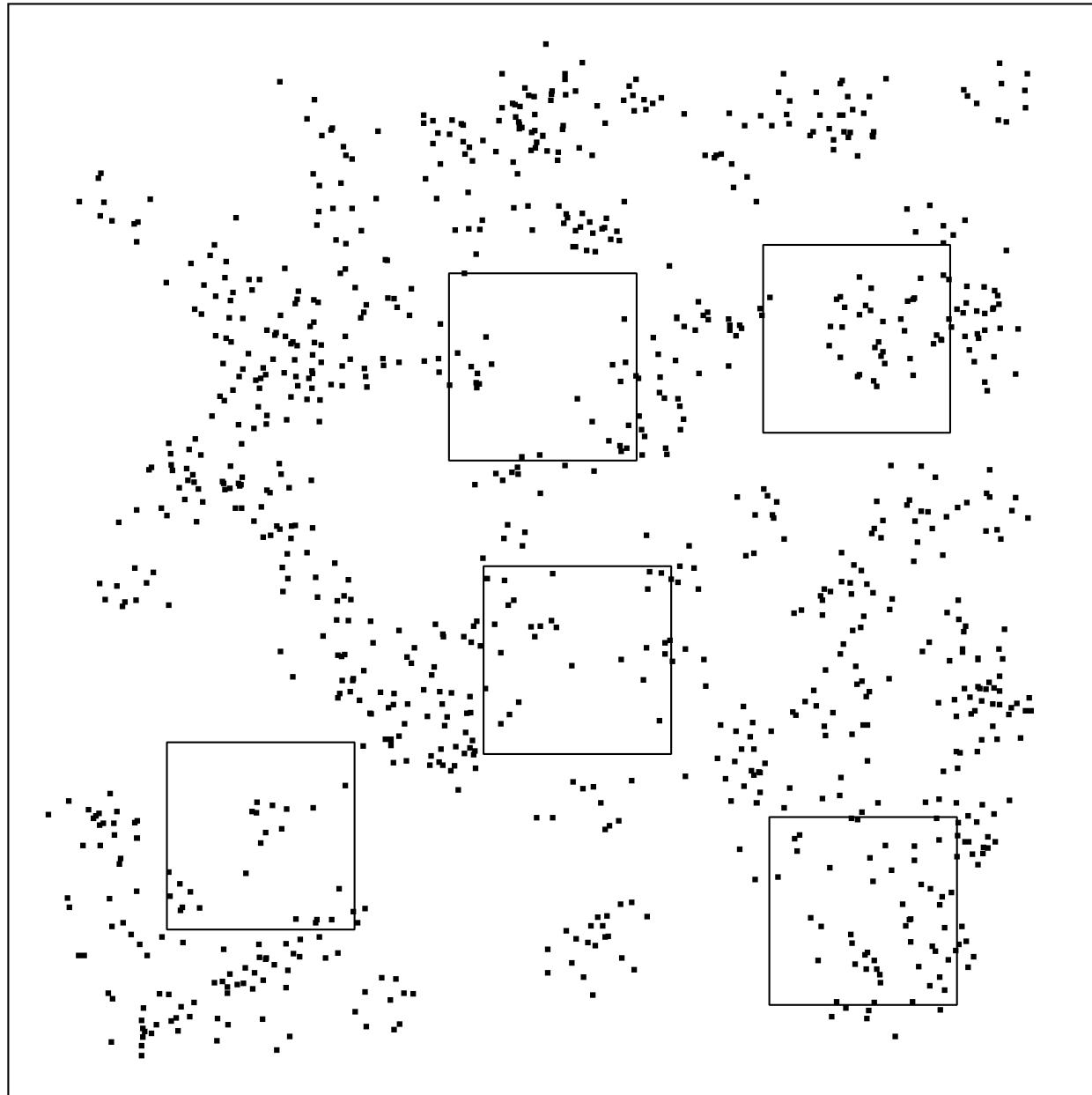
Sample - Count of sessile invertebrates

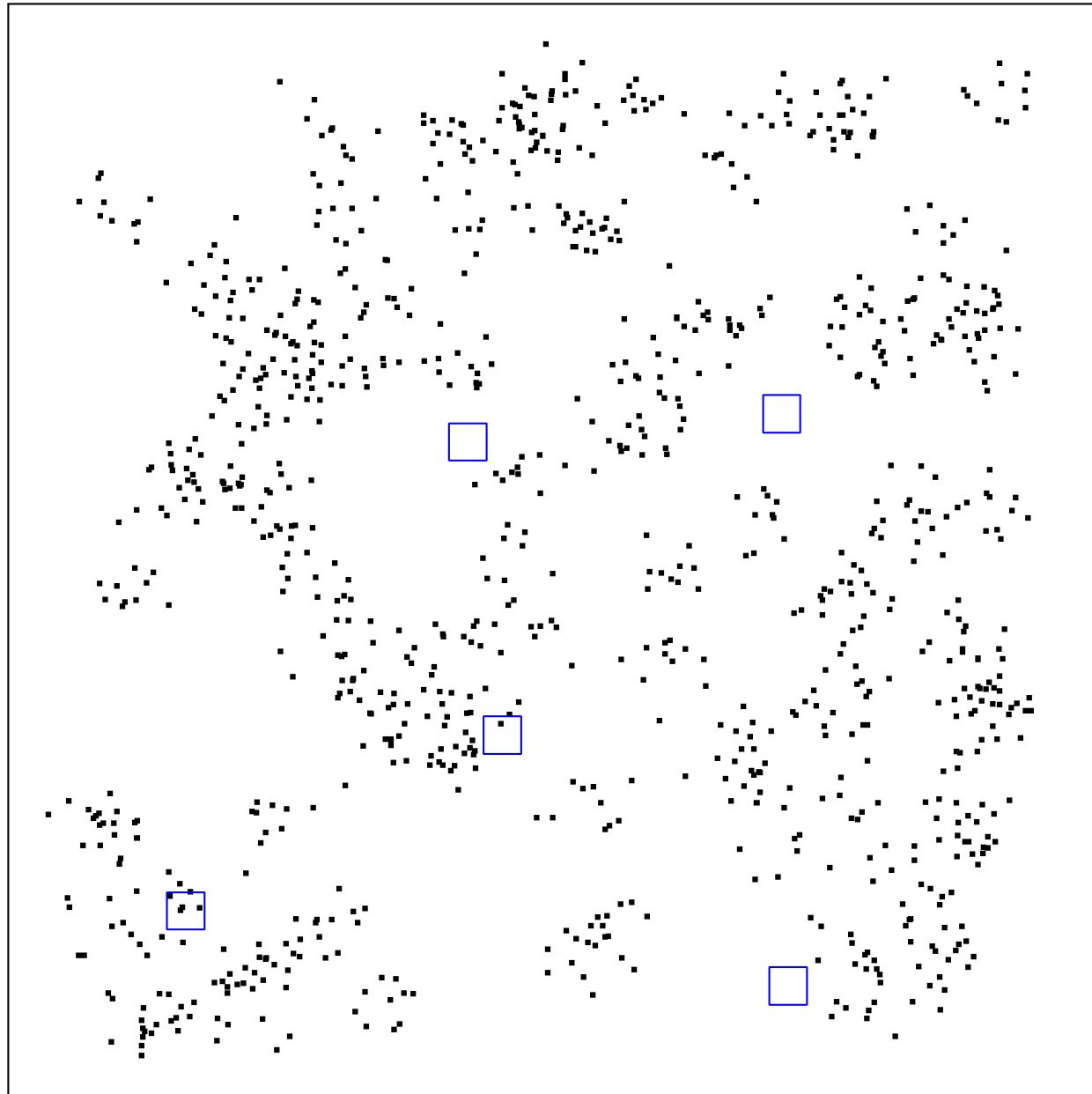
Population - The number of sessile invertebrates which settle on a 10 cm x 10 cm plate after 1 month

Statistic - Sample mean

Parameter - Population mean

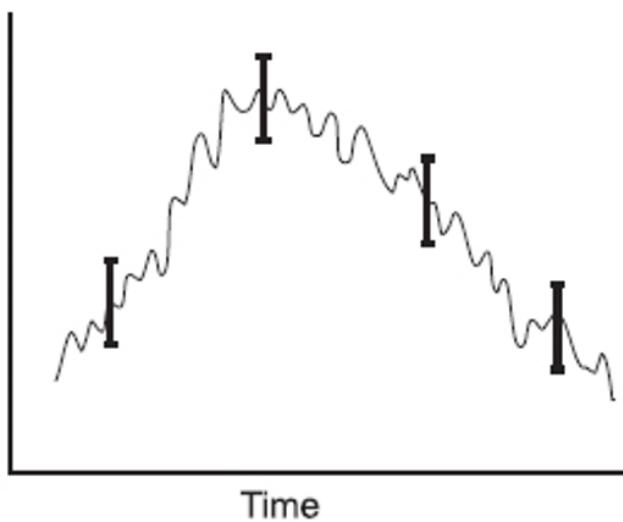




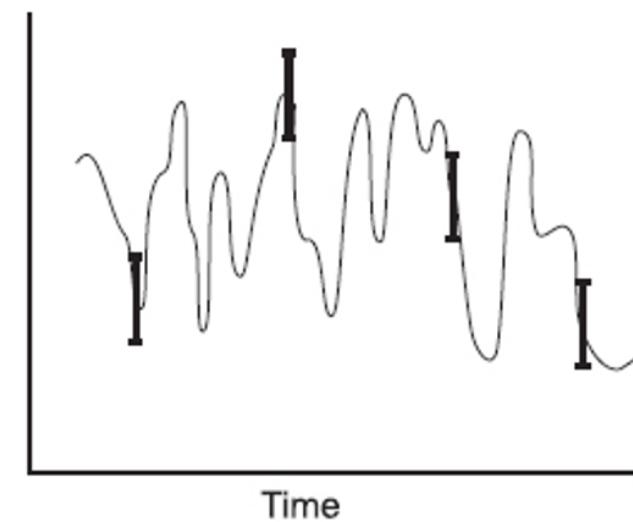


1.40

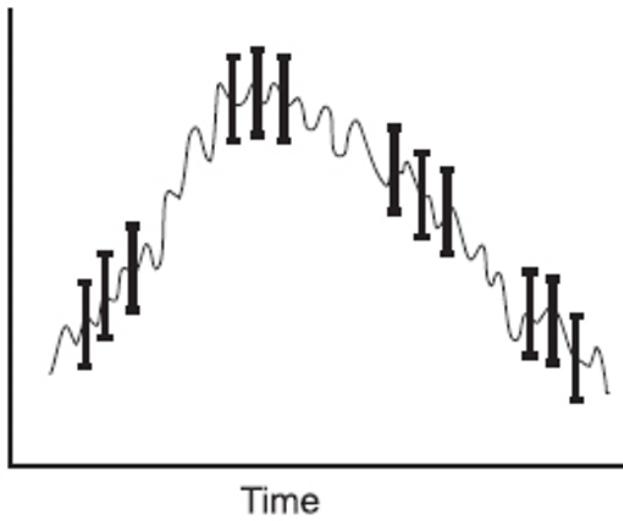
(a)



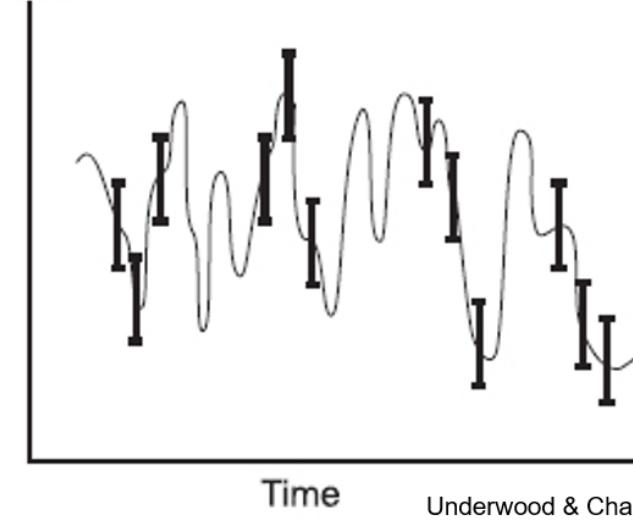
(b)



(c)



(d)



Underwood & Chapman (2013)

Randomisation

Randomisation ensures **independence** between sampling units, as well as that the sample you have is **representative** of the population of interest. This in turn ensures that our estimates of population parameters (means, treatment effects) are unbiased and our statistical inferences (conclusions from the statistical test) are reliable.

Random assignments

How do we ensure individuals allocated to treatments are independent and representative? Assign them randomly.

Random sampling

How do we ensure sampling units in space are independent and representative? Generate their positions randomly.

Simple random sample (SRS)

Simple random sampling is the most straightforward valid sampling method (accurate estimates of population parameters). In this method, each sampling unit within the population has an equal chance of being chosen (or assigned to each treatment). The downside is that it can lack precision as it doesn't take environmental variability into account.

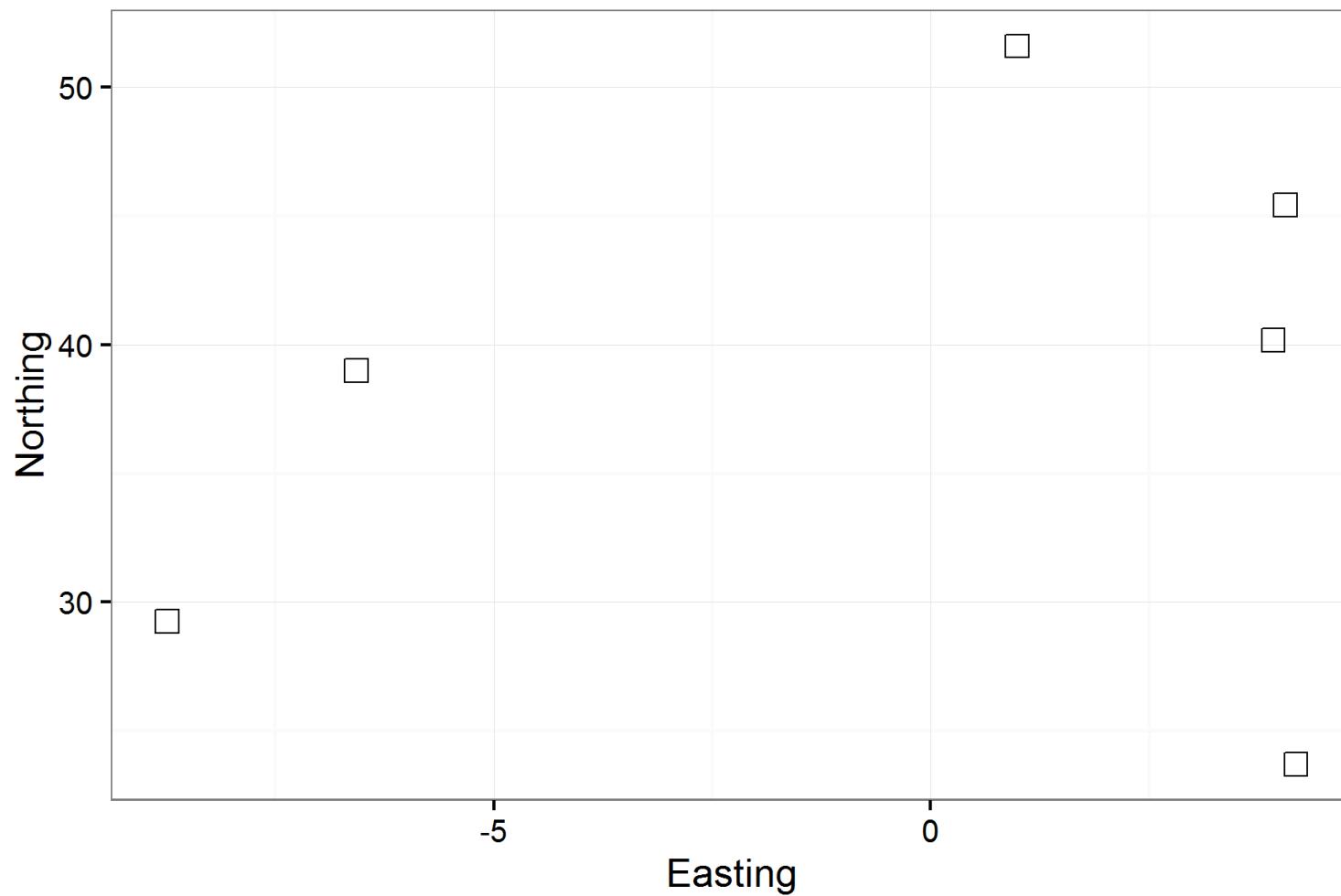
For spatial sampling, most commonly needed in ecology, the location of samples is generated randomly.

Sampling randomly in space

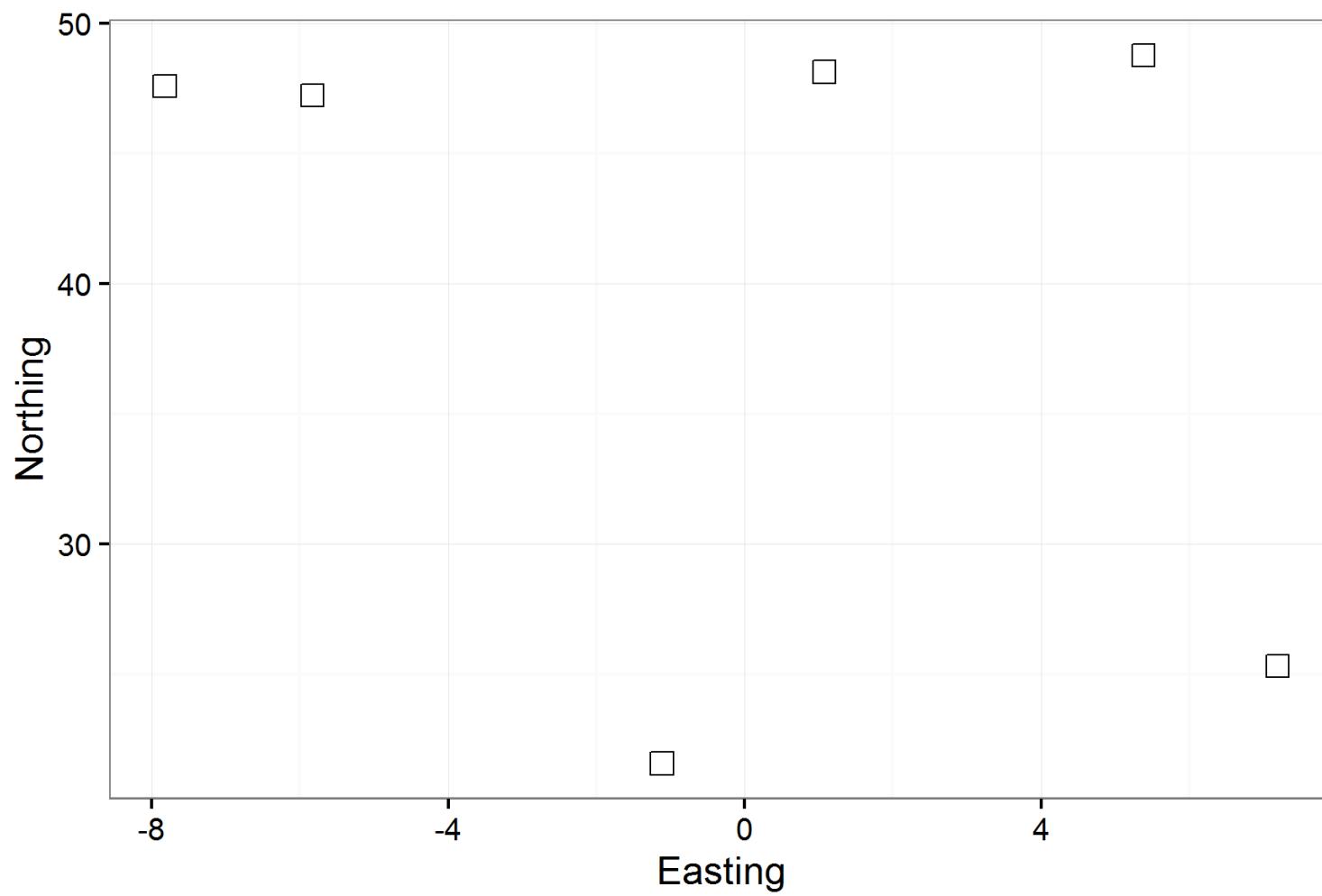
We use the `runif` function to sample randomly between minimum and maximum values.

```
> n=10 # sample size  
> minE=-12  
> maxE=8  
> minN=18  
> maxN=53  
> sample=data.frame(Easting=runif(n,minE,maxE),  
Northing=runif(n,minN,maxN))  
> head(sample)
```

	Easting	Northing
1	-3.3221228	40.32446
2	7.3056080	28.82126
3	3.8157806	31.28448
4	-2.6834025	33.29521
5	4.5332681	32.00745
6	0.9729939	48.42175



1.45



1.46

Random allocation of treatment

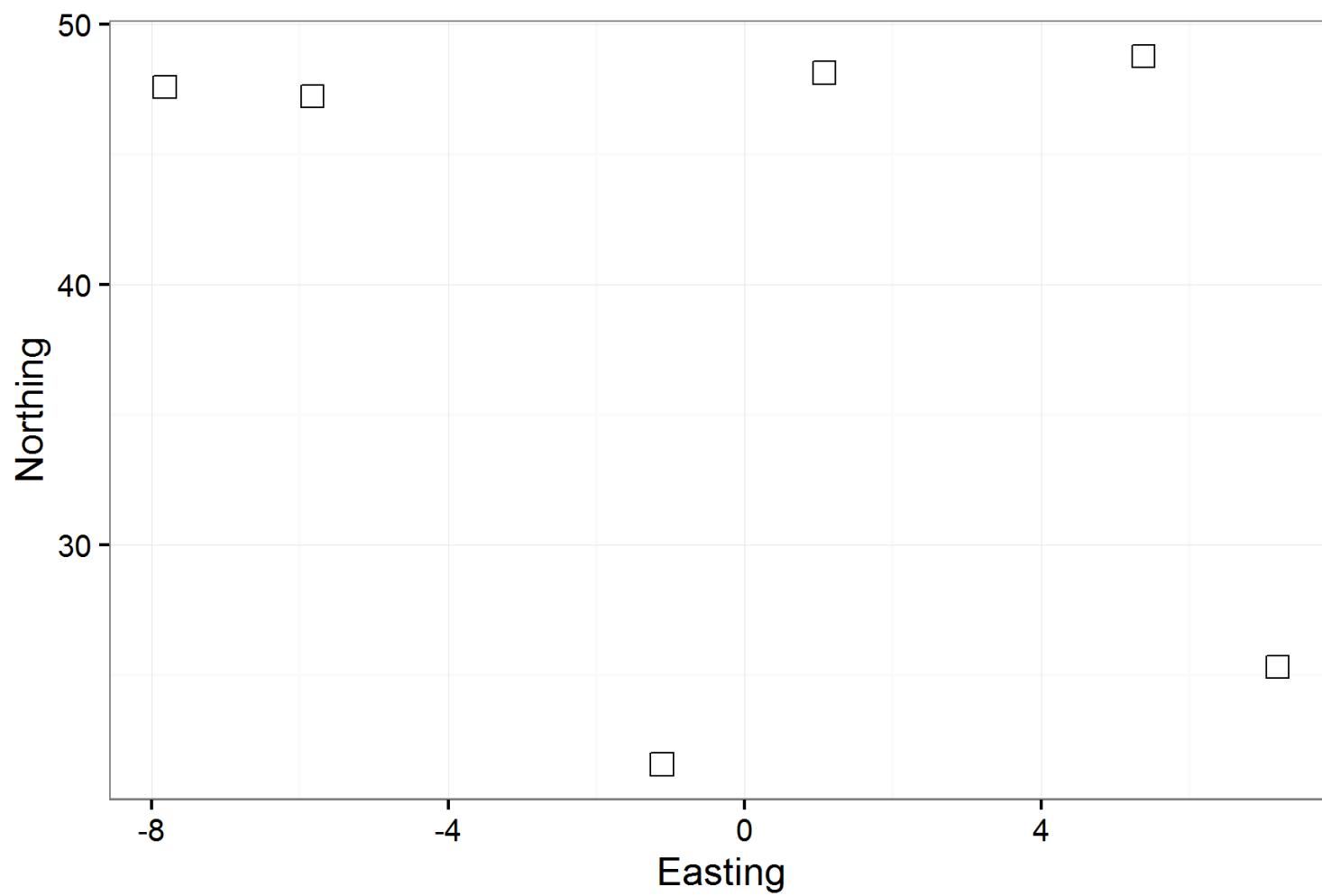
We use the `sample` function to randomise treatments to subjects. Here we have 9 subjects and 3 treatments.

```
> N=9 # sample size  
> subject=1:N  
> treatment=rep(c("T1","T2","T3"),each=3)  
> treatment=sample(treatment)  
> sample=data.frame(subject,treatment)  
> head(sample)
```

	subject	treatment
1	1	T3
2	2	T2
3	3	T2
4	4	T1
5	5	T3
6	6	T3

Haphazard samples

Haphazard samples are samples which aim to be random, but the placement of samples is done less formally than truly random samples. For example, locations are chosen by throwing an object over your shoulder to create a sample you intend to be random. In practice your choices can be influenced by factors you aren't aware of. For example, you might subconsciously exclude an item and include another because you know one item would be easier to locate than another. Its not possible to identify and eliminate all your biases, which is why random sampling is preferred.



1.49

Stratified sampling

Stratification aims to improve precision while maintaining accuracy. Using simple random sampling, we may sample within some strata only rarely, and therefore not be able to have precise estimates of population parameters within these strata. This is particularly true if strata are very small.

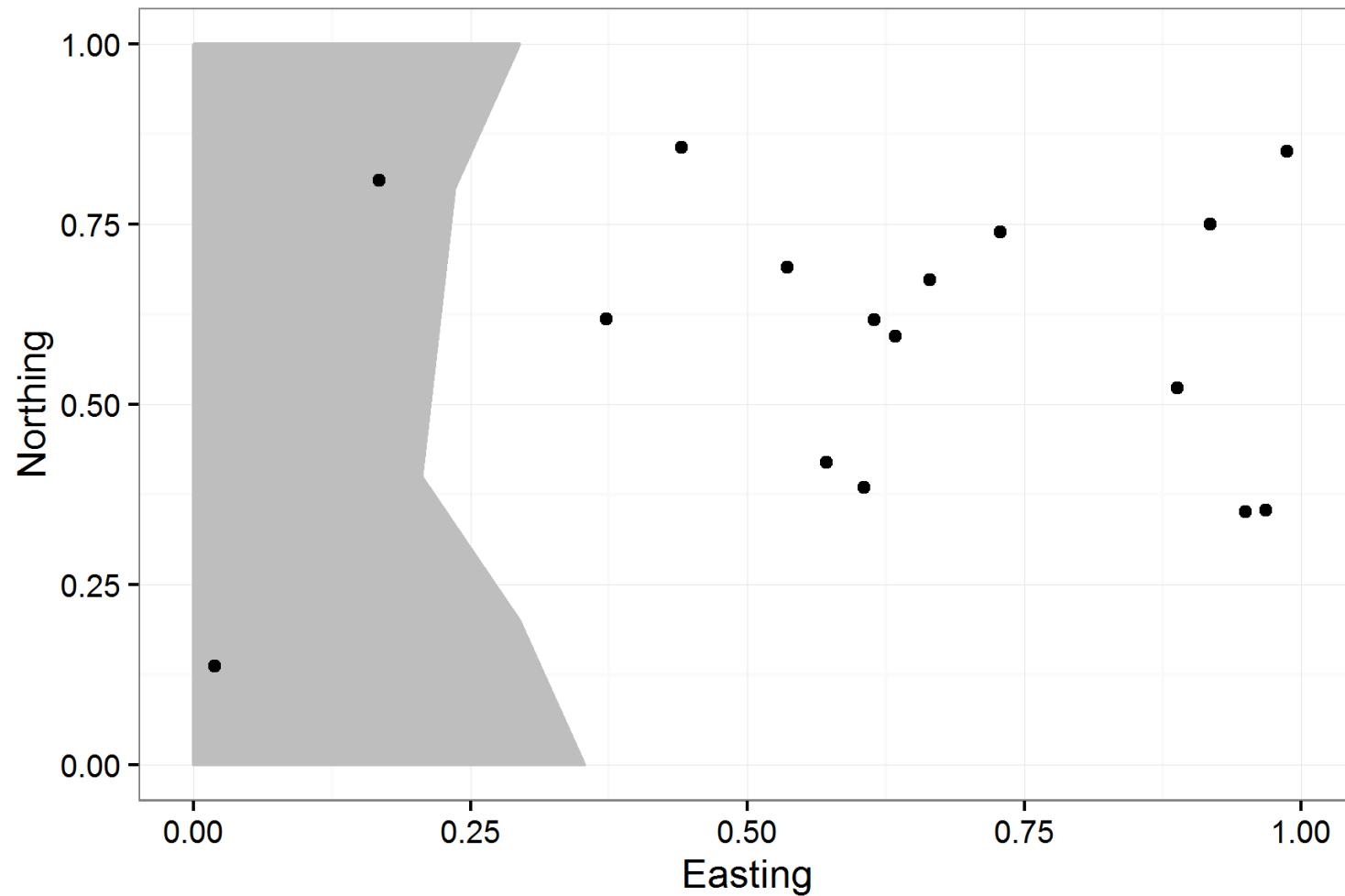
- Divide population into strata which represent clearly defined units
- Sample independently (SRS) within each stratum.
- Include strata as predictor variables in analysis

A stratified sampling design is more difficult to design and analyse, and can only be applied when there are mutually exclusive and well defined strata.

Example

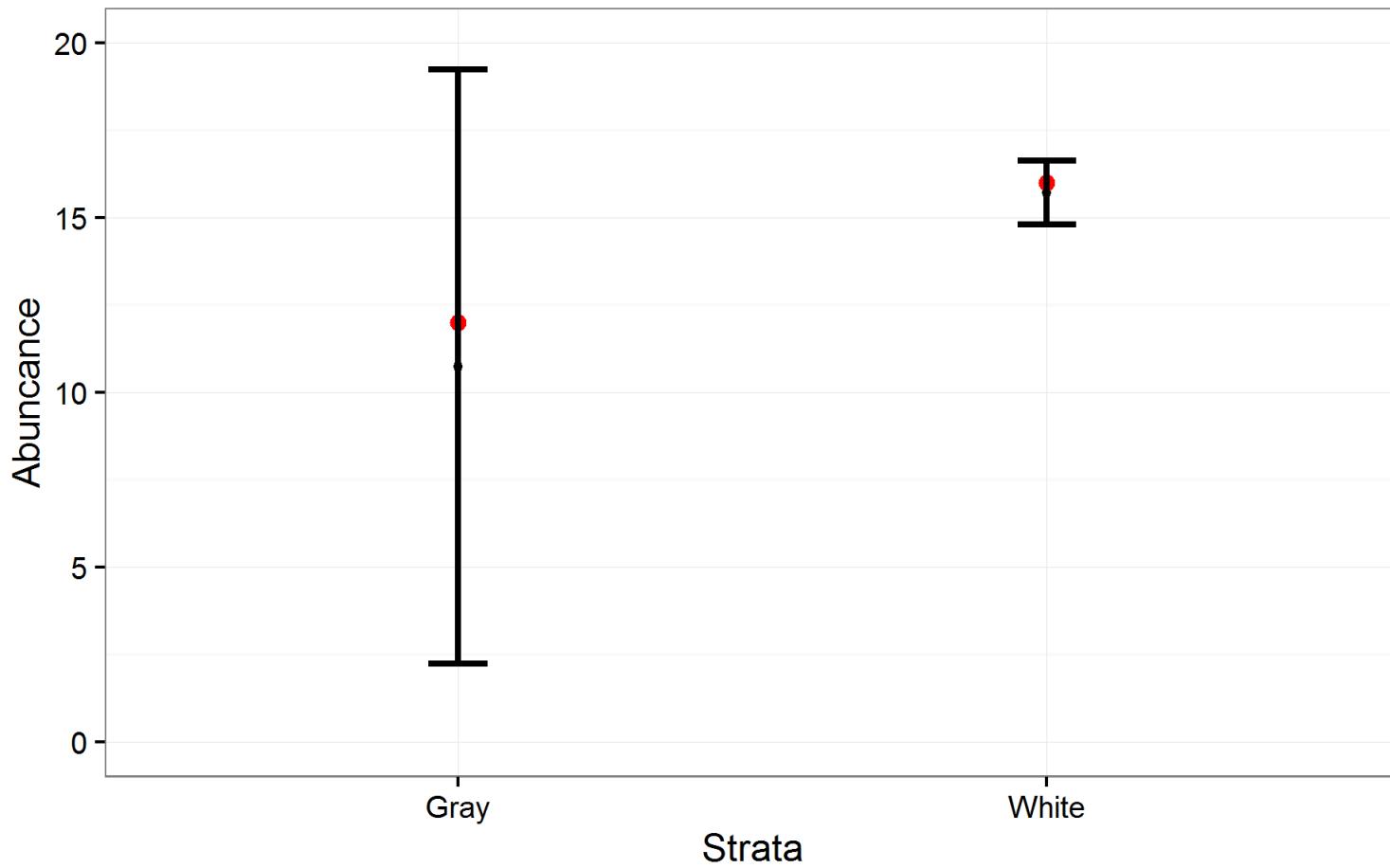
We want to compare species richness between urban and remote estuaries. Estuaries have a number of distinct environments including rocky shore and mangroves.

Random sampling, no stratification



1.51

Random sampling, no stratification



1.52

We need to decide what proportion of the sample to sample within each stratum. Options include:

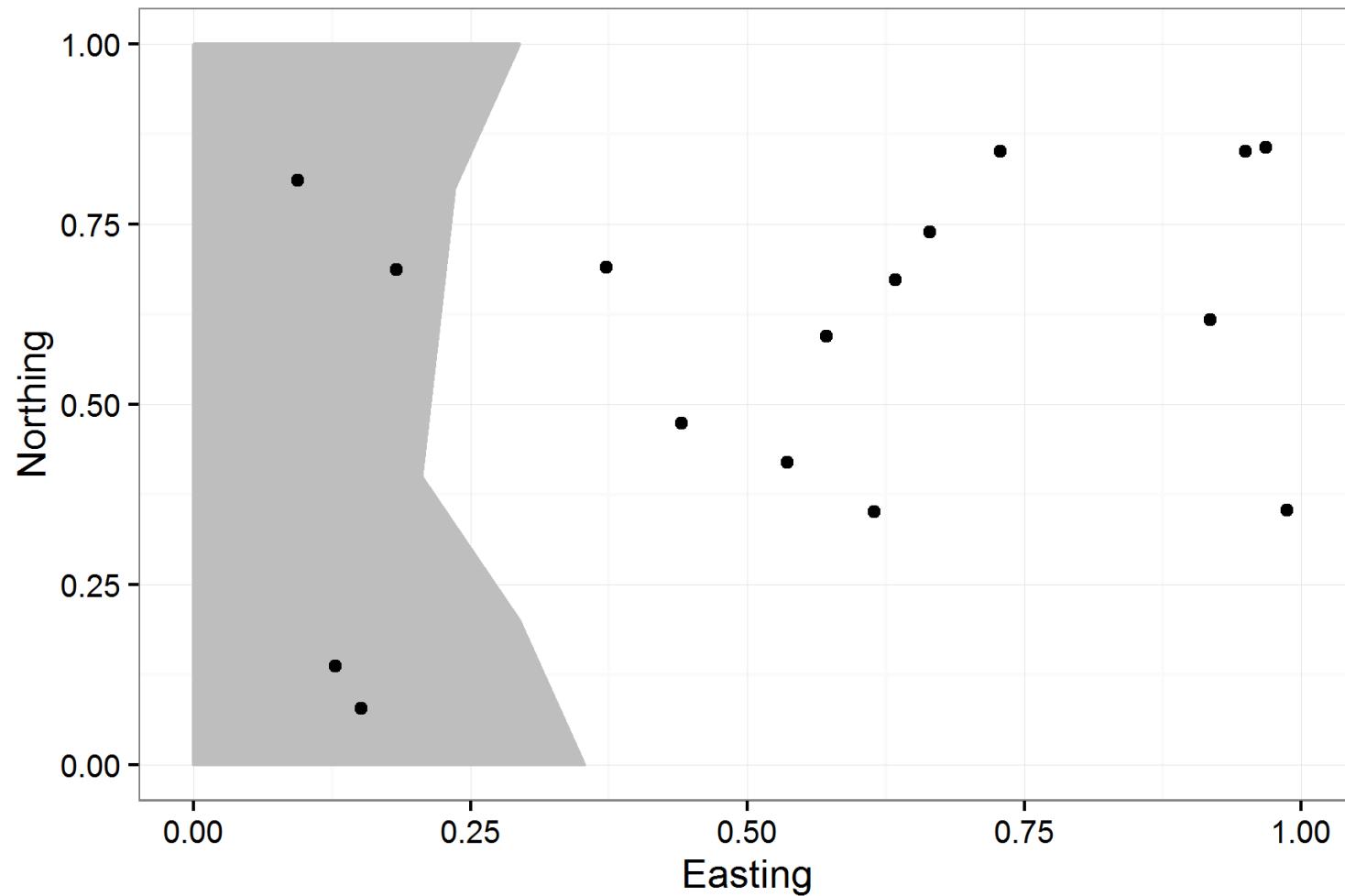
Proportional

- The number of samples is proportional to the area of the strata
- Ensures all strata have a representative number of samples, and no strata is unstapled
- Valid estimate of all population parameters

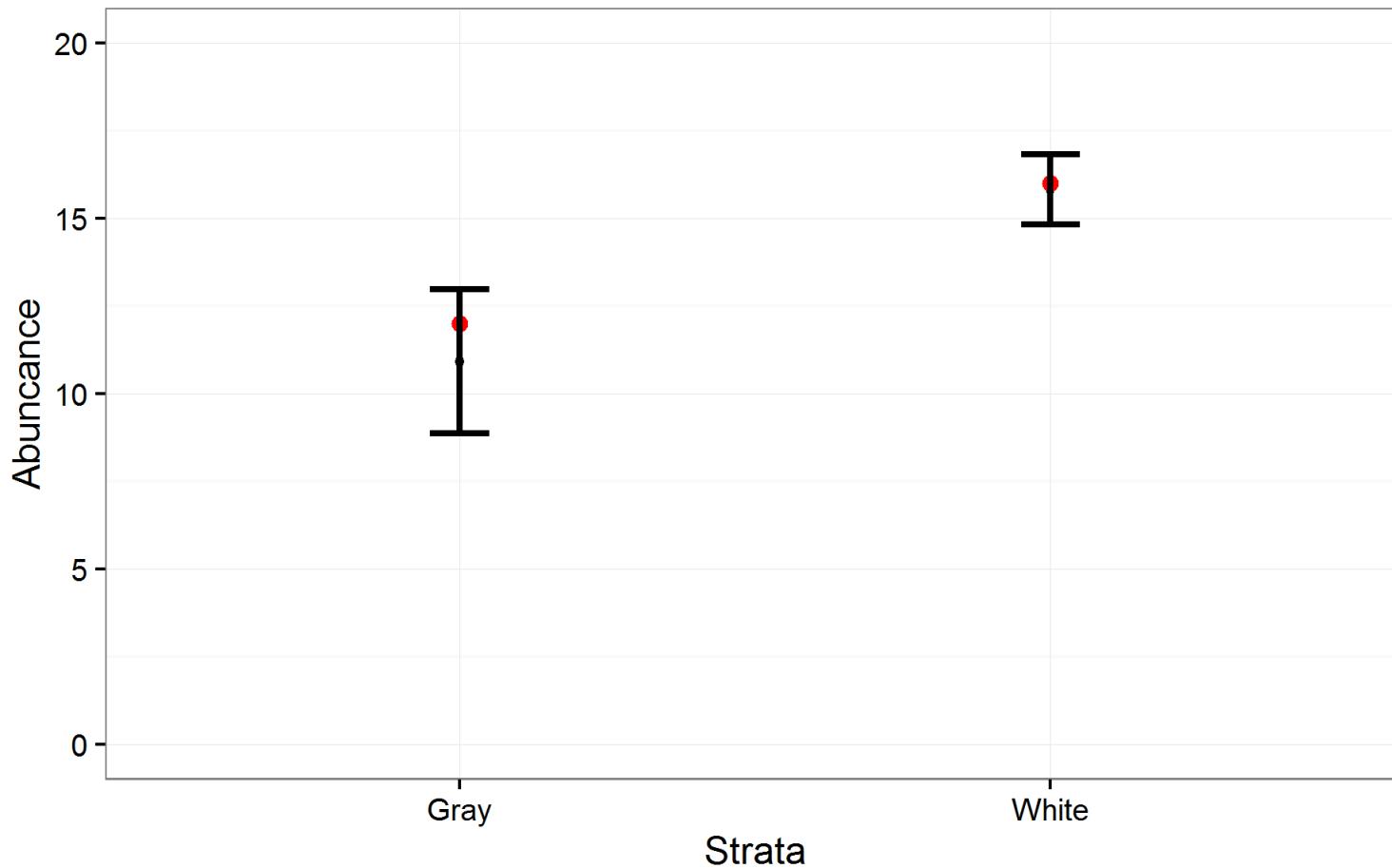
Equal

- The number of samples is equal in all strata
- Ensures no strata is unsampled
- May improve estimates for smaller strata
- Valid estimates of population parameters can be obtained, but stratification must be taken into account

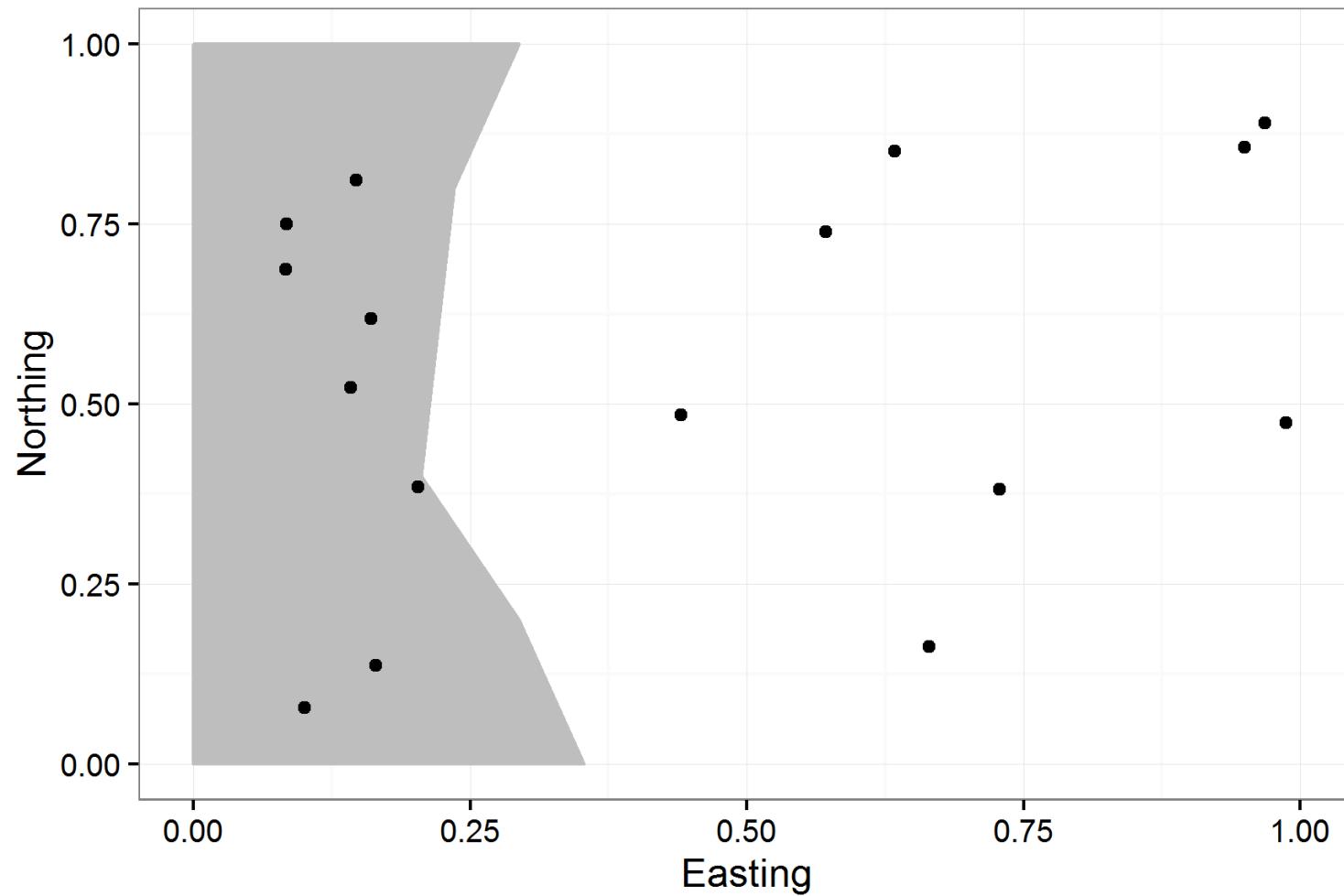
Stratified sampling, sample size proportional to size of strata



Stratified sampling, sample size proportional to size of strata

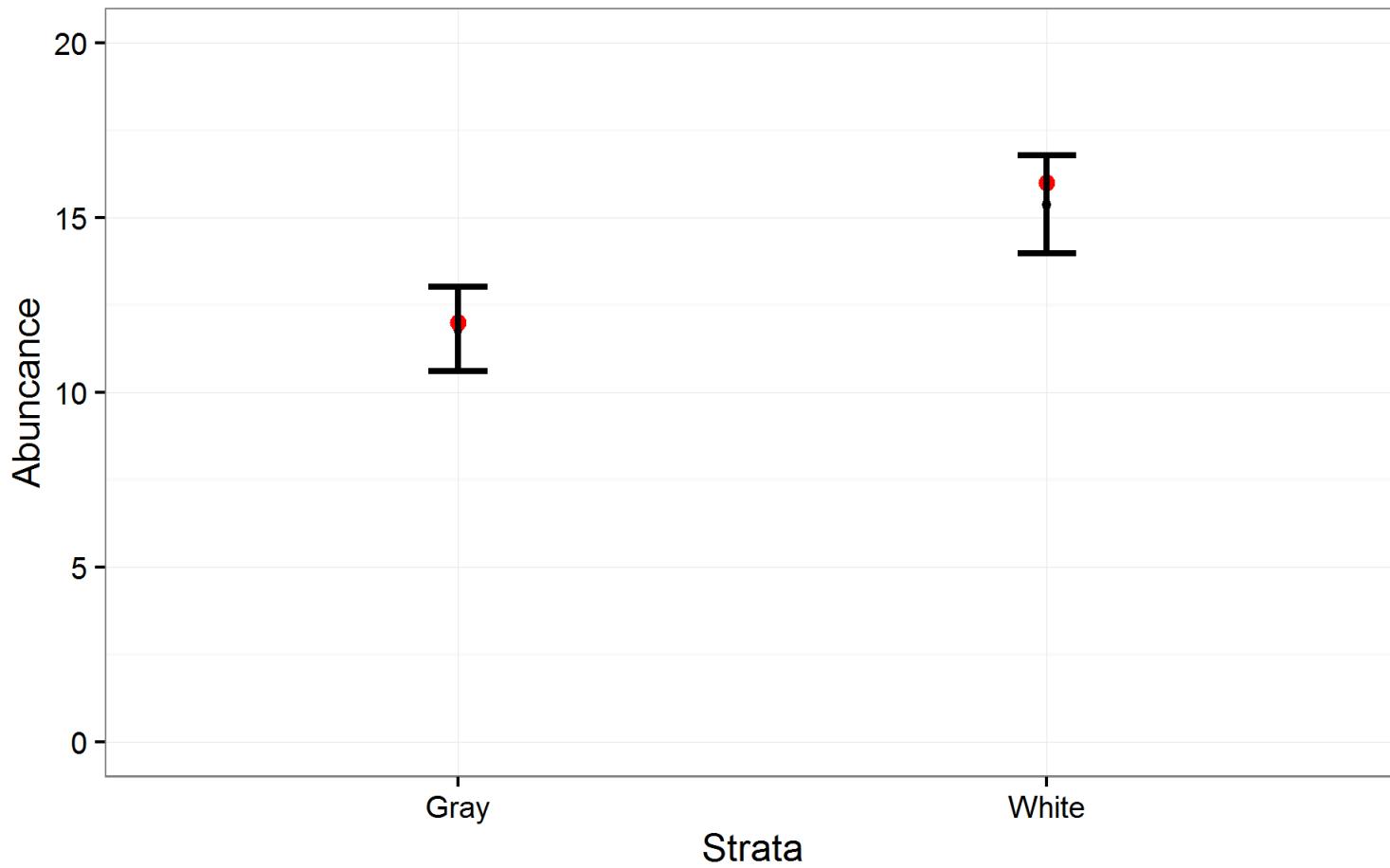


Stratified sampling, sample size equal in each stratum



1.56

Stratified sampling, sample size equal in each stratum



Optimal sample size

The strategy with the most bang for your buck is to sample most in strata that are most variable. This requires some information about the variability within strata, which you may have from say a pilot study.

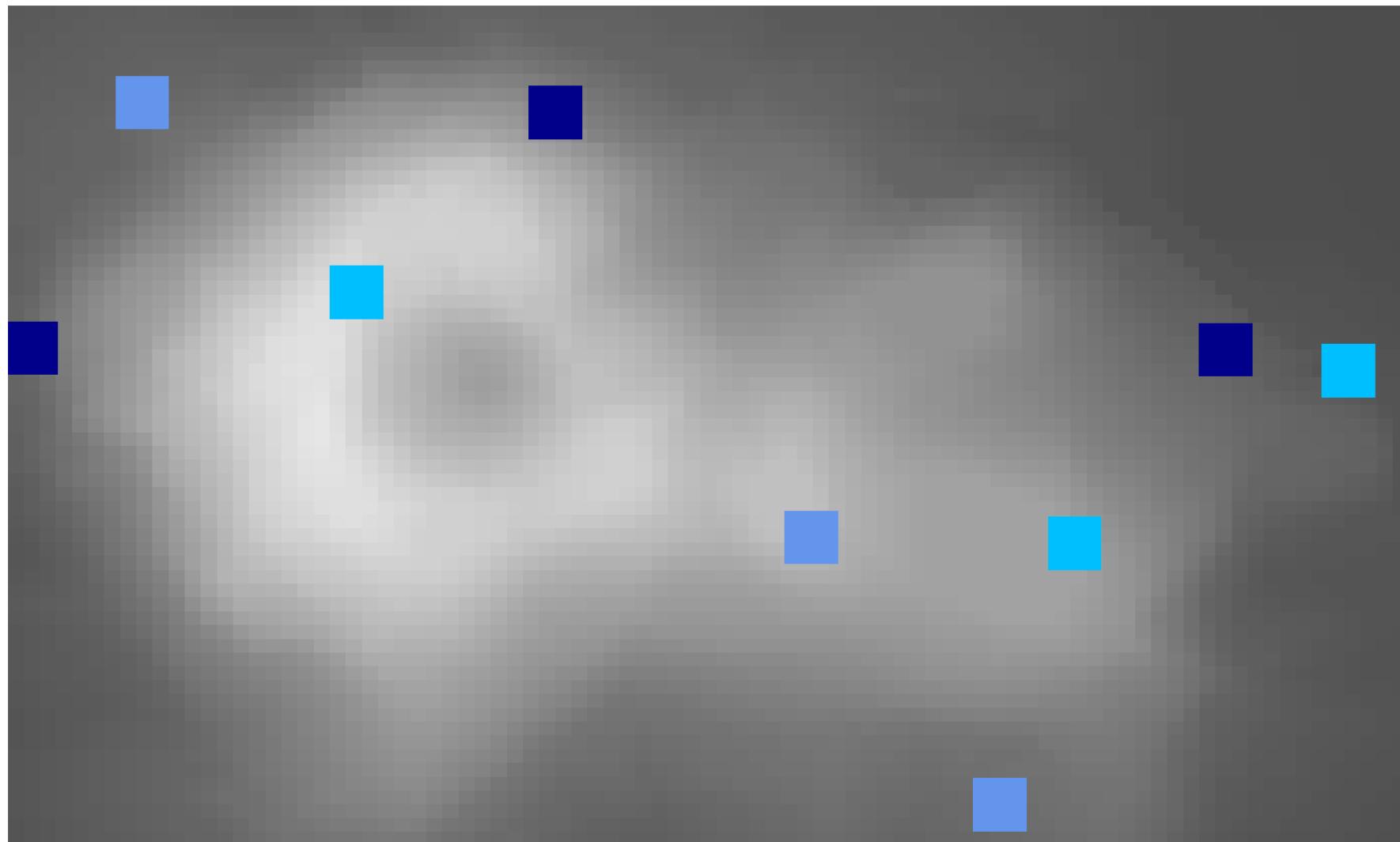
Blocking

Random allocation of experimental units can lead to high levels of variation between units, which may obscure the effects of the treatment factor of interest. Grouping units into blocks with similar conditions (in space or time) can explain some variation, this can lead to more precise estimates of parameters of interest and more powerful tests.

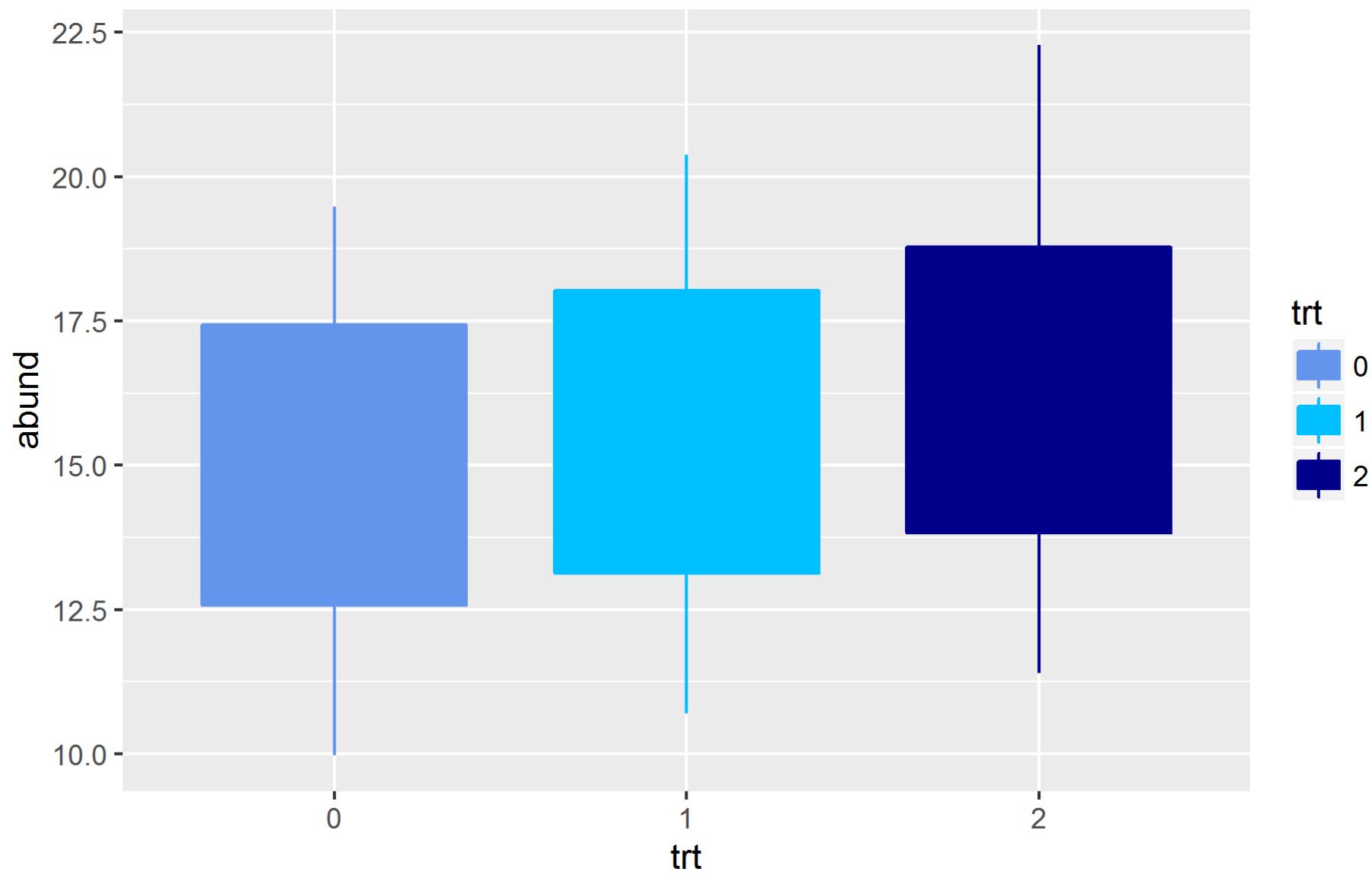
Blocking does not have to be in space or time, blocks can be formed from other attributes like size or age of organisms or some other attribute like aridity.

Example

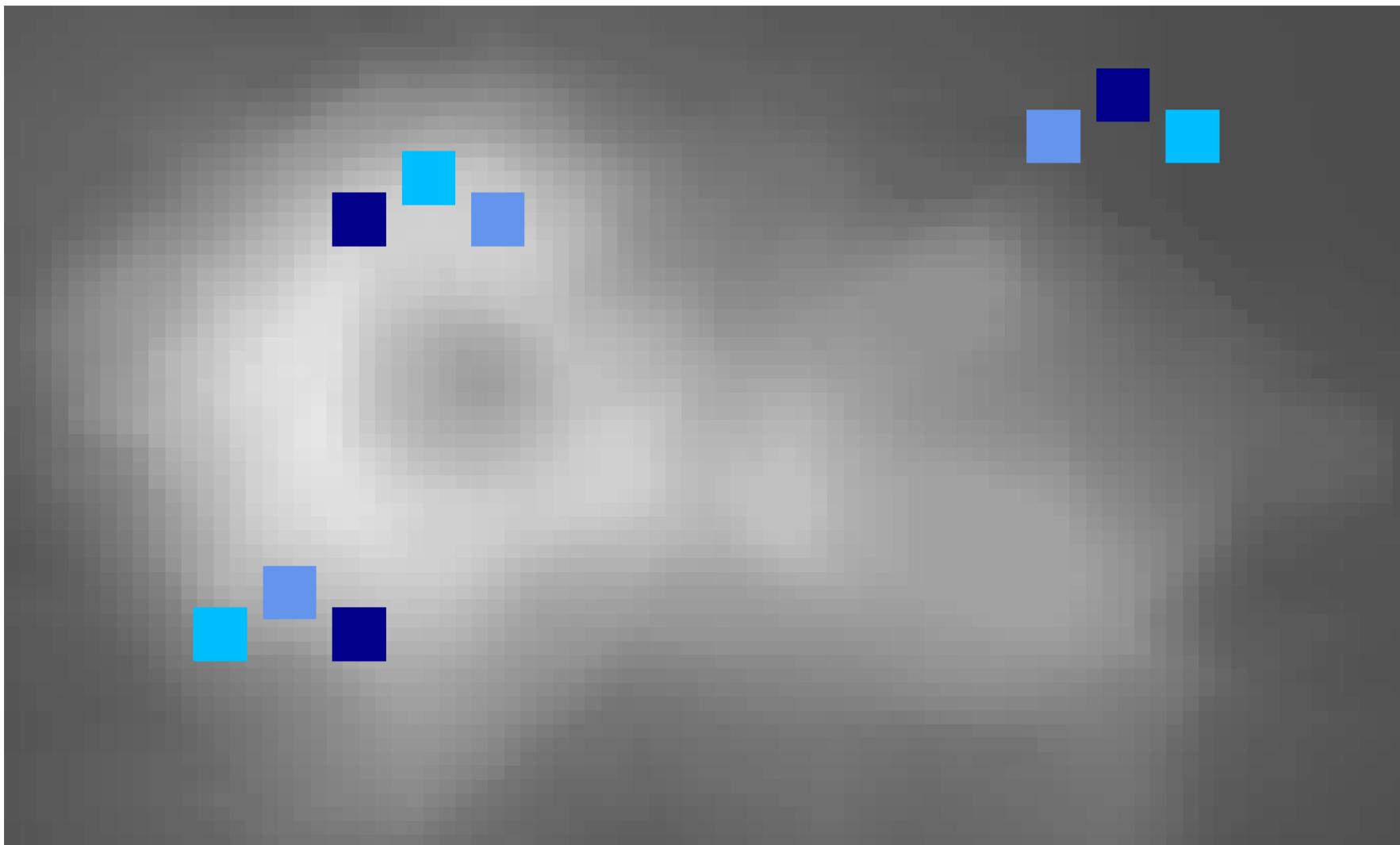
Faeth (1992) applied one of four leaf damage treatments to four branches within eight randomly chosen trees of an evergreen oak. Trees were blocks and leaf damage was the treatment. Each damage treatment was represented once (on a single branch, the experimental unit) in each block.



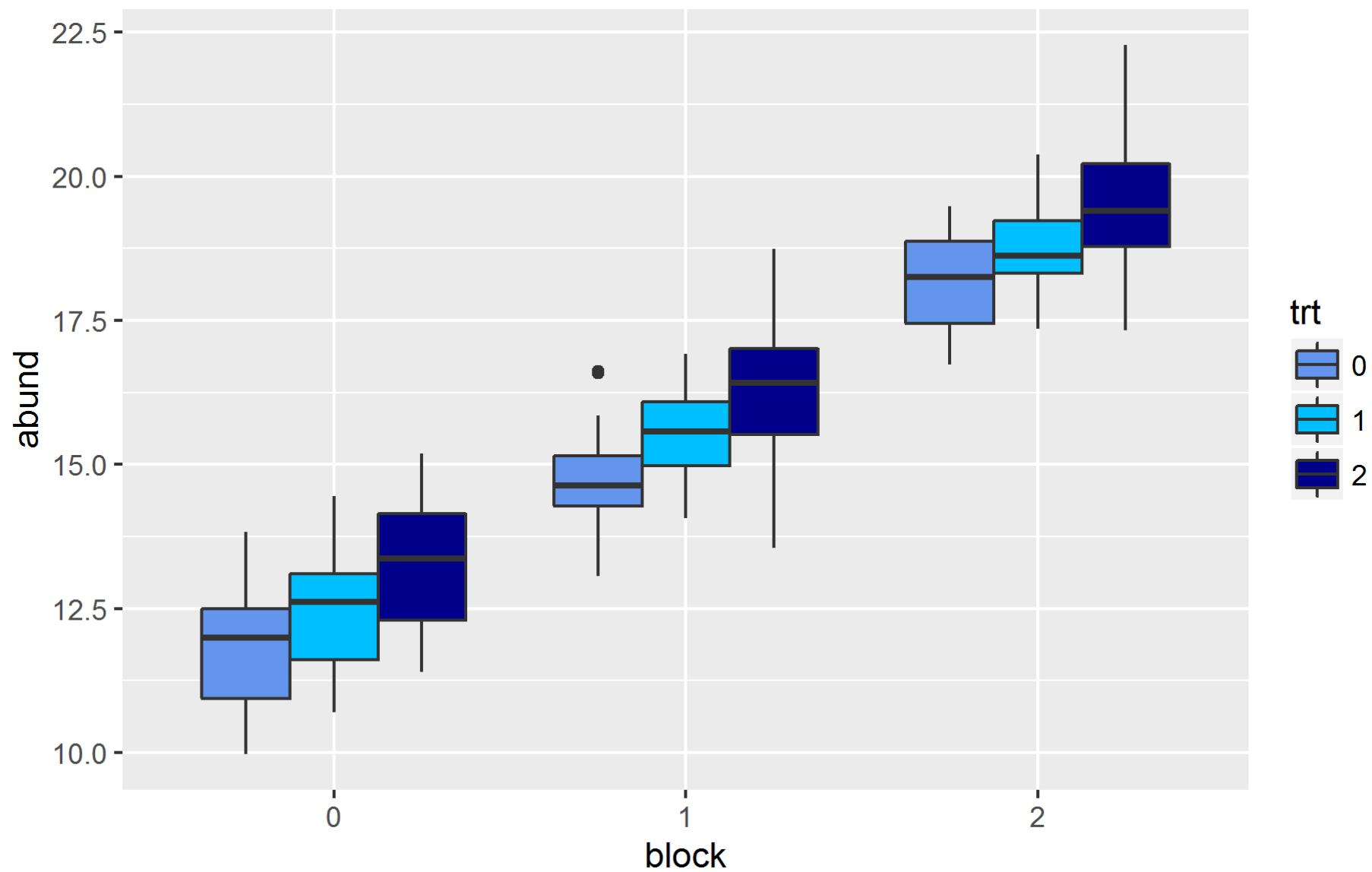
1.60



1.61



1.62

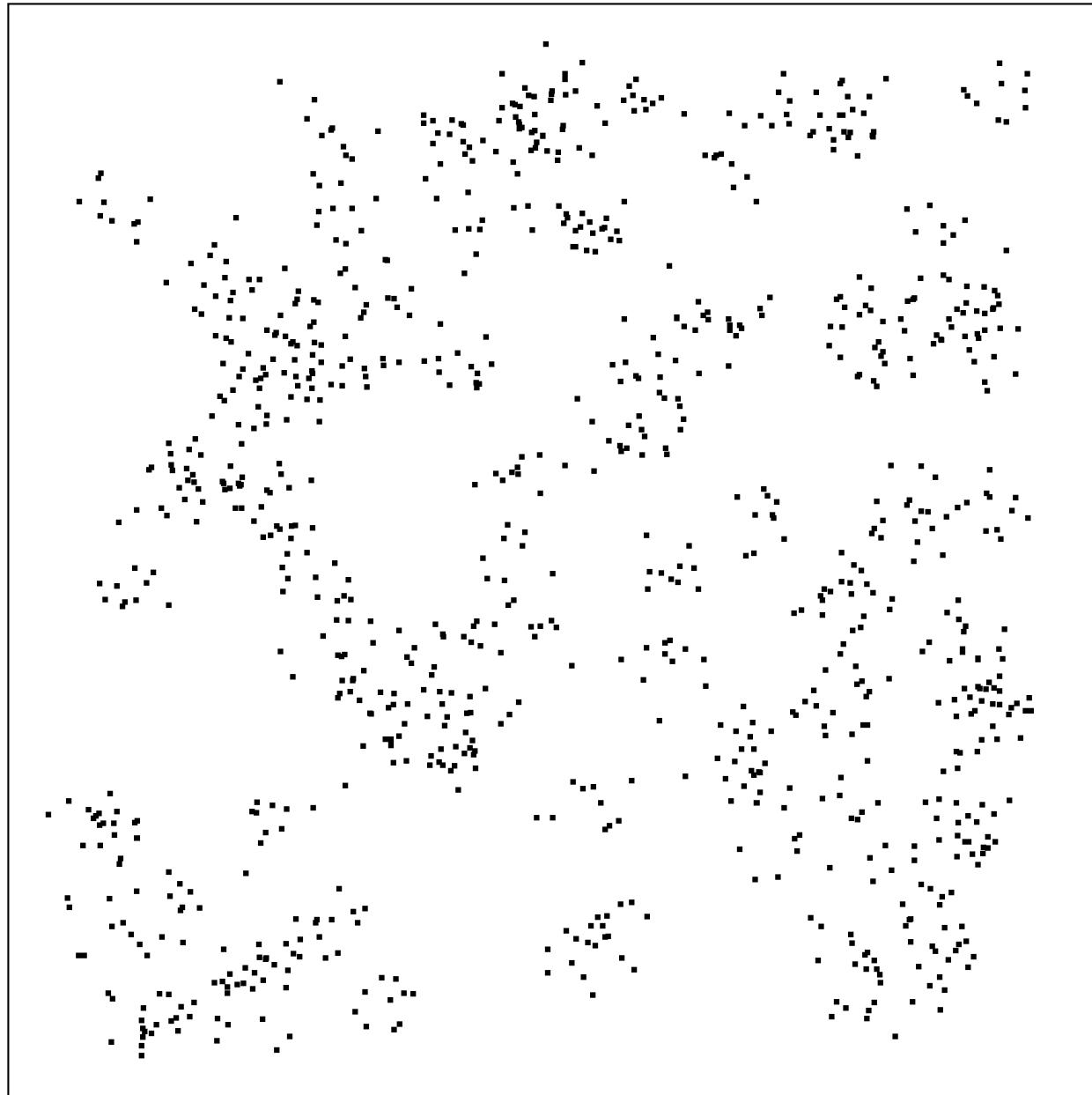


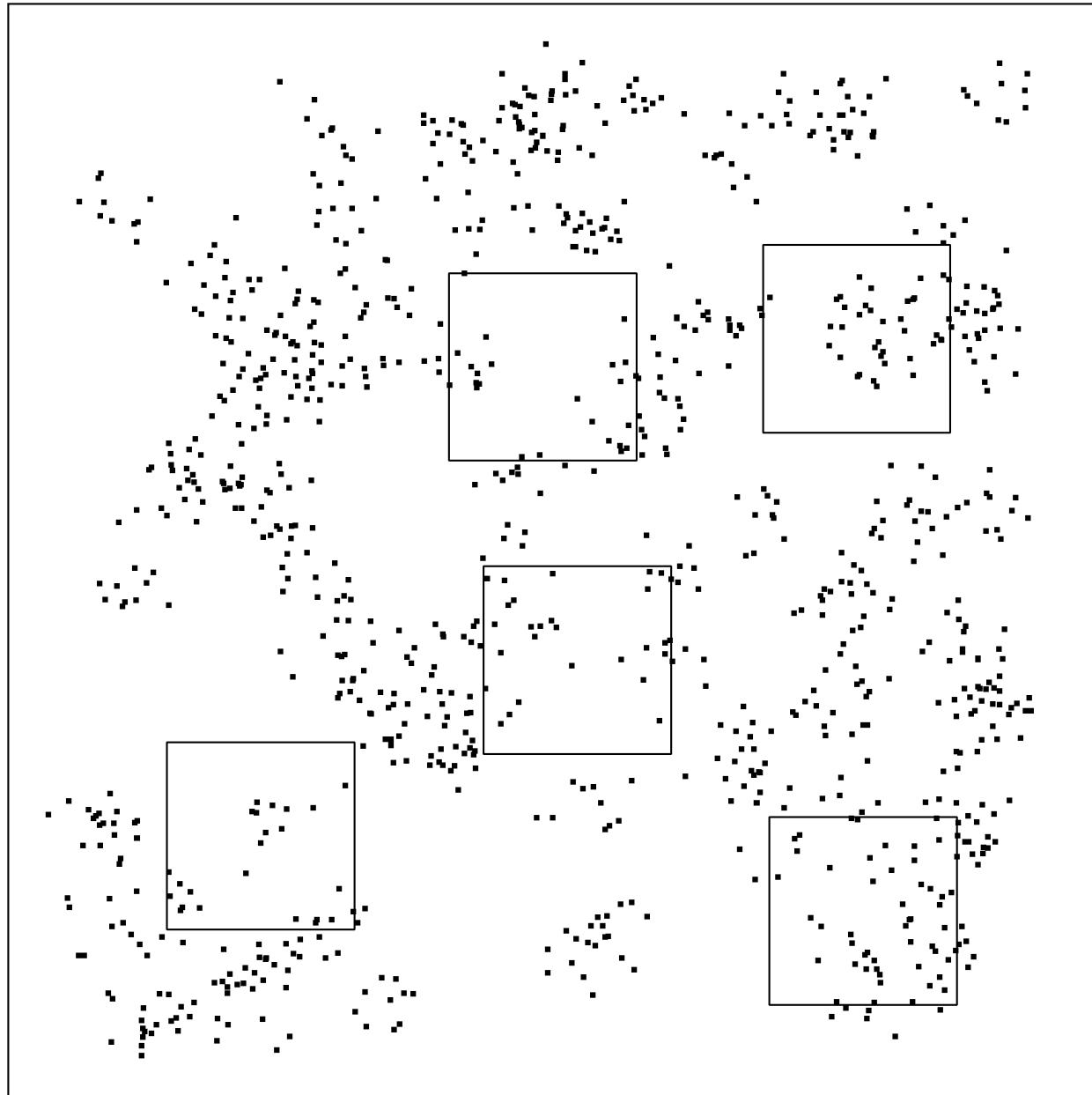
1.63

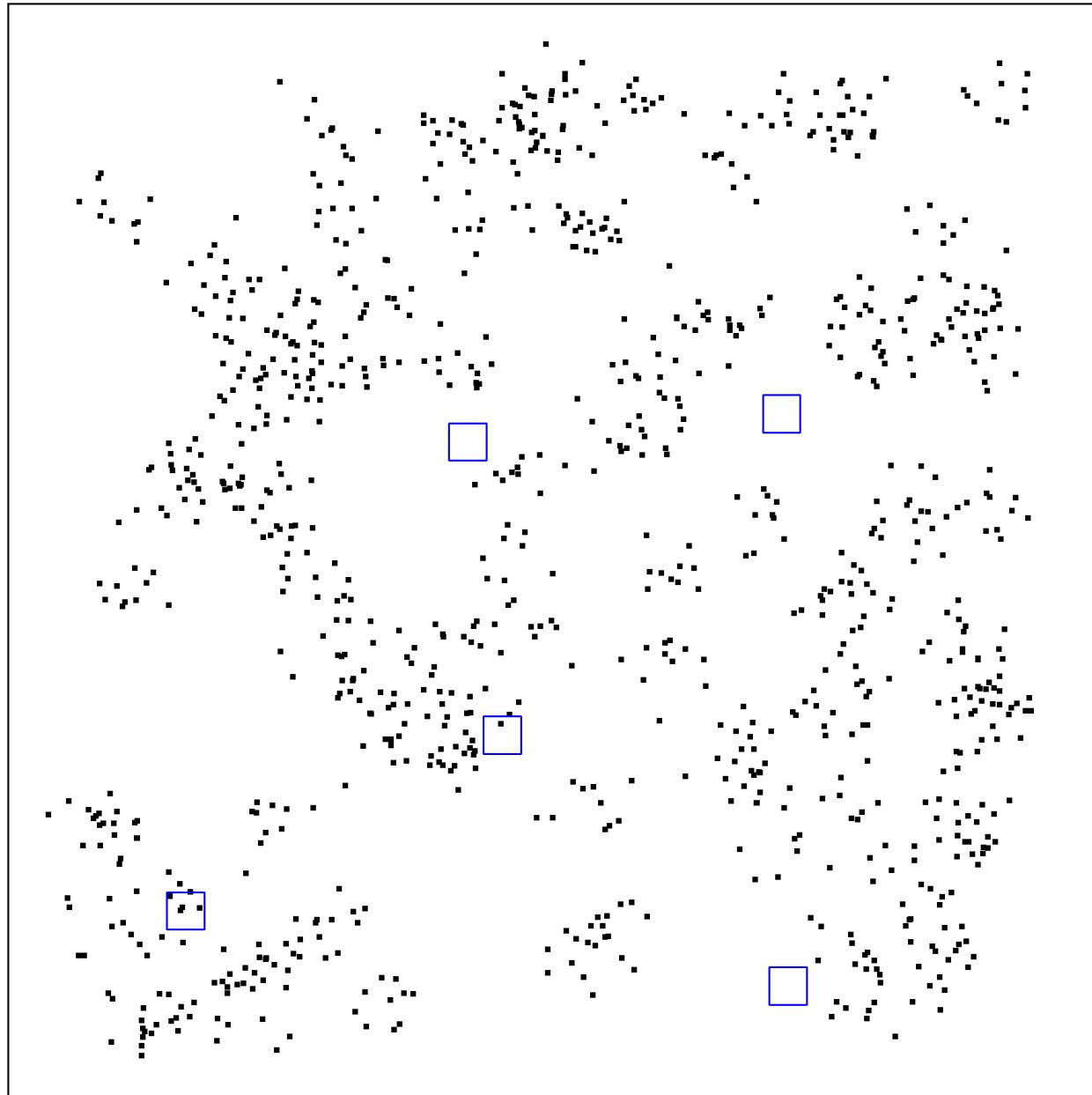
Blocking needs to be accounted for in analysis, and can be very advantageous if there is a lot of variation between blocks. However, unnecessary blocking (where there is minimal or no variation between blocks) can lead to poorer power. Pilot studies are the best way to estimate this variation and plan the most powerful experimental design.

Hierarchical blocks

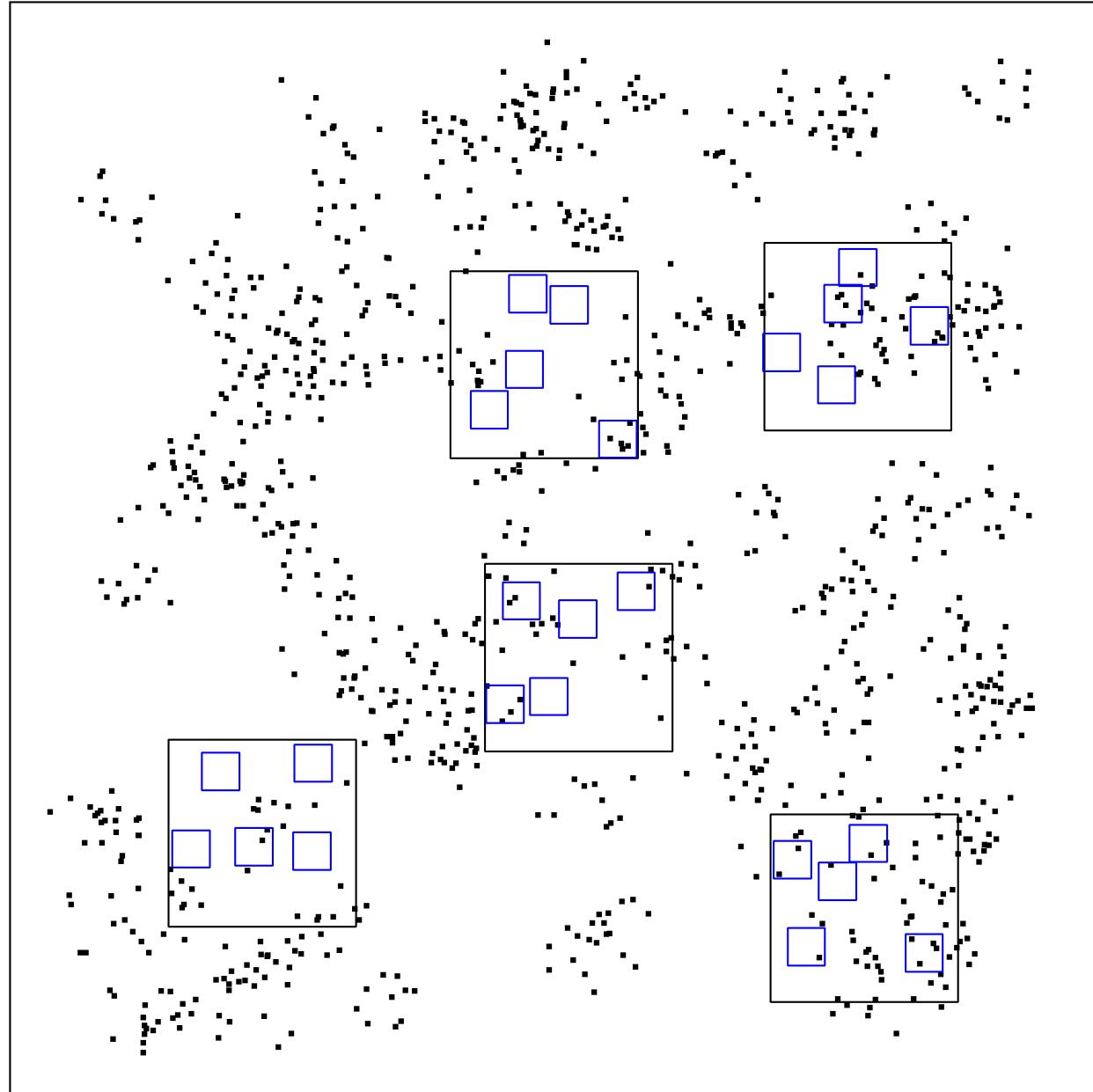
Blocking can be done hierarchically to further reduce variation, as well as to estimate variation at different scales, and is often a good idea in ecology.







1.68

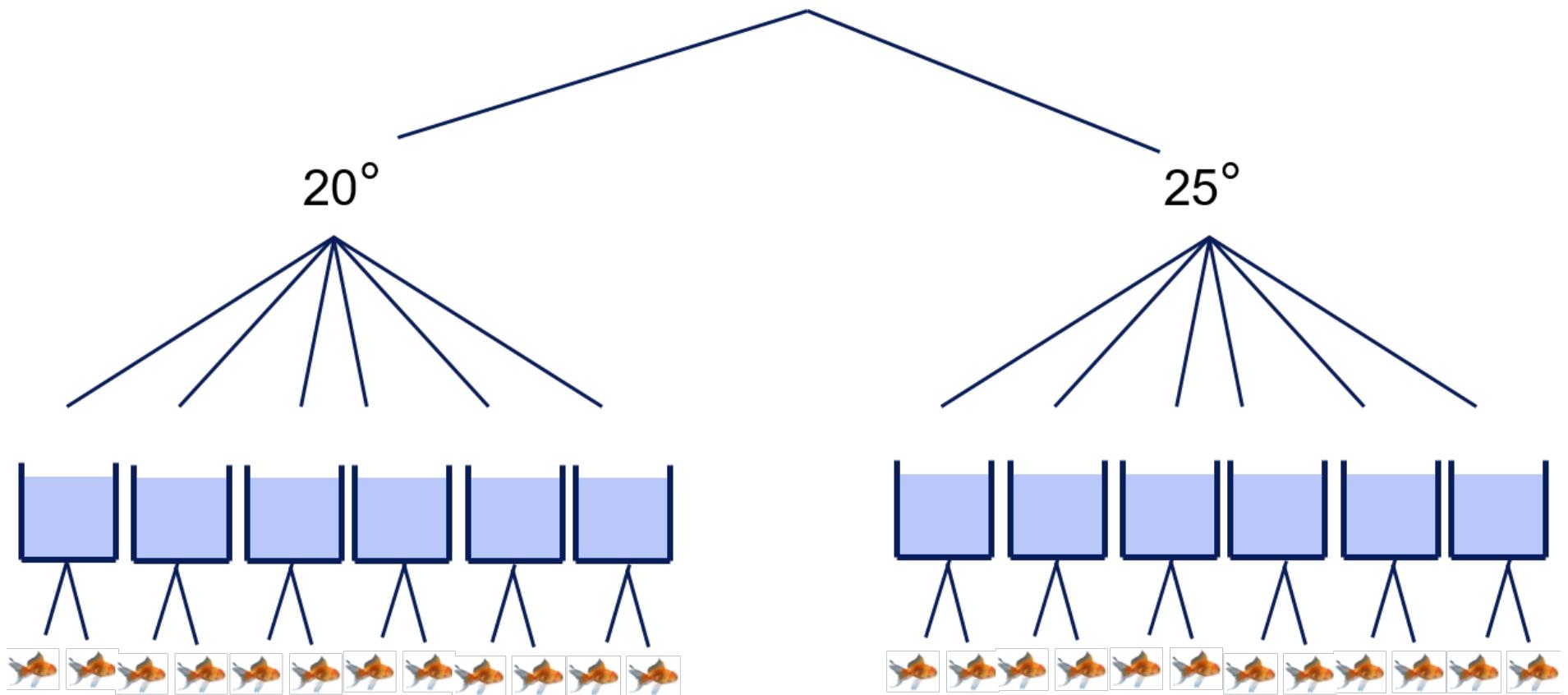


1.69

Now we can investigate variation at two spatial scales. From this sample the standard deviation within blocks on the smaller scale is about 1.5, so quite high, which makes sense, as many of the smaller blocks have nothing in them. If we then average the smaller blocks to estimate the abundance of the bigger blocks, we find the standard deviation on the scale of the big blocks is about 0.6. This makes sense, because the bigger blocks tend to contain some areas with lots of individuals, and some areas with very few, so on average they are not that variable, while the small blocks tend to either be in an area with individuals, or without any, so they are more variable.

By clustering our sampling in this way we have found out a lot more about the scales on which this species varies. However the price we pay is that analysis is more complicated, this sampling structure needs to be accounted for in any analysis.

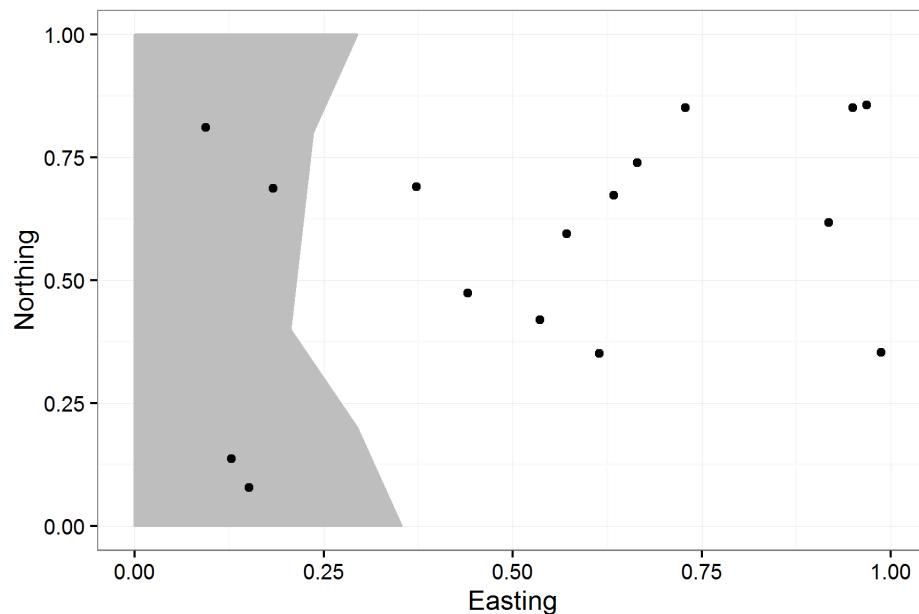
Which units are independent/dependent?



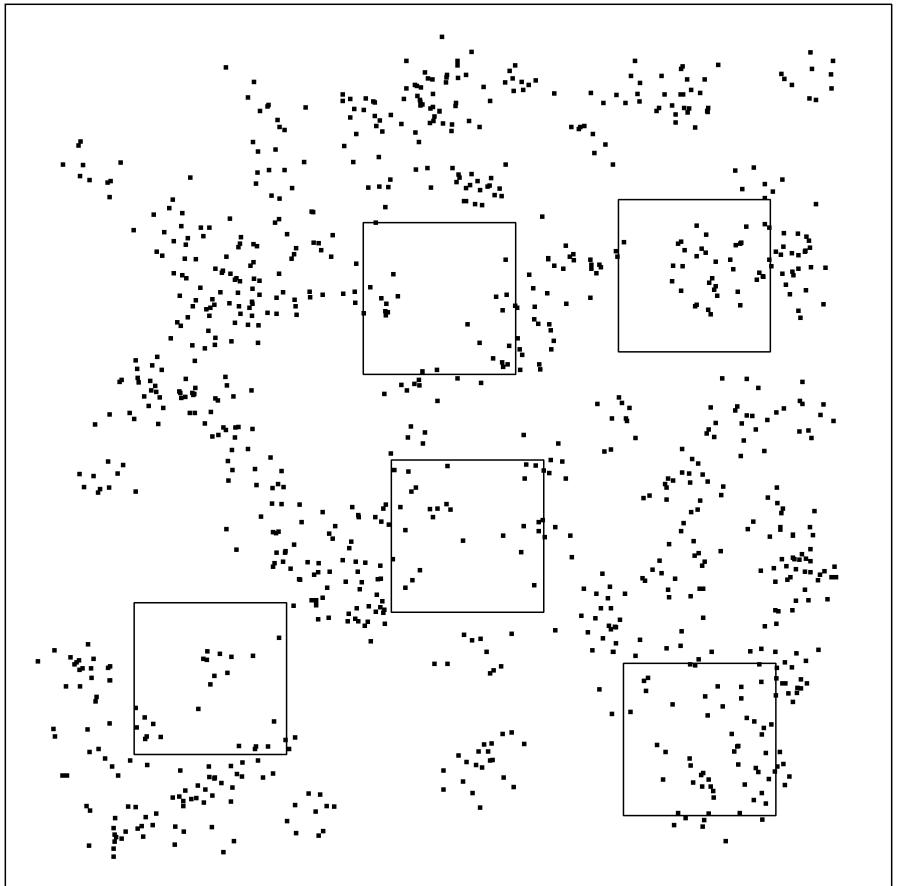
For the fish we can go one step further, and estimate variation within tanks.

Random and fixed blocks (strata)

Blocking can be based on random or fixed variables. This has consequences for how you sample, as well as analysis.



In our first example, we stratified according to environment, with two environmental types represented. There are only two environmental types, and we are interested in them specifically. These are fixed blocks, and we treat them as fixed in any analyses.



In this example we chose 5 blocks from a larger possible population of blocks (all locations in space). These 5 blocks are not anything special, they are just chosen at random, and the inference we are interested in is not specific to these blocks, but for the whole population of possible blocks. These are random blocks, and must be treated as such in any analysis. In addition we have to make sure we really do sample them randomly.

Pilot studies

Designing experiments most efficiently requires some knowledge about the variability in the system being studied.

Pilot studies are small scale experiments which are designed to help in the planning of larger experiments. Some advantages of pilot studies include trying out experimental practices to see how well they work and gathering some information on variability. They are excellent practice and can save a lot of time and money later on. All the sampling designs to follow can be informed to a greater or lesser extent by pilot studies.

Types of studies

There are broadly speaking two types of studies in ecology, manipulative and observational.

In **manipulative experiments** the treatment is imposed (randomly) to sampling units while in **observational studies** the variable of interest is sampled at different levels present in the environment.

Manipulative experiments can derive **causal relationships** while observational studies cannot. This is because there may be confounding variables for which we have not accounted. The process of randomly assigning treatments to sampling units in a manipulative experiment controls for confounding, assuming the controls are well designed.

We tested how temperature affects fish by putting fish in tanks with different temperatures.

What kind of study is this?

We assessed the impact of building a golf course by testing species abundance at affected and unaffected sites.

What kind of study is this? Could the effect be due to something else other than the golf course?

Power and sample size

Once we have an experimental design which has all the properties of good design, a clear question, controls, replication, independence, blocking, we still need to decide on a sample size. The sample size required depends on **effect size** and **variation**, as well as the sampling design and intended analysis.

The aim of an experiment is to detect an effect if one is present. Failure to do so is called type II error.

	H_0 True	H_0 False
Reject H_0	Type I Error	Correct Rejection
Fail to Reject H_0	Correct Decision	Type II Error

Sample size demonstration

http://onlinestatbook.com/stat_sim/sampling_dist/

There is no point running an experiment if you have very little chance of detecting an effect. If we have some information about effect size and variability we can conduct a power analysis before starting an experiment, to make sure our sample size is sufficient.

The best way to estimate **variability** is to conduct a pilot study. If that is not possible, you can often find estimates of variability in previous literature.

Effect size

This is the magnitude of the effect you are hoping to detect. For a simple analysis with two samples (e.g. treatment and control) it is the difference between the population means of these groups.

We do not know the true effect size, that is why we are running the experiment. To chose an effect size for the power analysis, we need to consider what effect size is **ecologically meaningful**.

For example, we don't really care if the treatment and control groups differ by one or two, as this is not ecologically interesting, but if they differ by 3 we want to make sure we pick that up. So the effect size we use for the power analysis is 3.

Another common method is to look in previous literature for an effect size, however this may result in an effect size below what we consider relevant. A smaller effect size requires a larger sample size to achieve good power, and your power analysis might encourage you to sample more than is viable and cost effective.

Example

Katie wants to know if oysters filter water. She will set up tanks with and without oysters, and measure the light penetration after two hours. She knows from a pilot study that the standard deviation in light penetration in her water samples is 30%. She is interested in detecting a 20% difference between treatments. How many tanks does she need to achieve good power?

```
> library(pwr)
> sigma=30 # standard deviation within groups
> trt.effect=20 # ecologically meaningful difference between groups
> pwr.t.test(n = NULL, d=trt.effect/sigma, sig.level = 0.05, power=0.80)
```

Two-sample t test power calculation

n = 36.30566

d = 0.6666667

sig.level = 0.05

power = 0.8

alternative = two.sided

NOTE: n is number in *each* group

Intro Stats Revision

- Which method do you use when?
- Statistical inference
- Two-sample t -test
- Transformation

Which method do you use when?

When thinking about how to analyse data, there are two key things to think about:

- **What is the research question?**
- What are the main properties of my data?

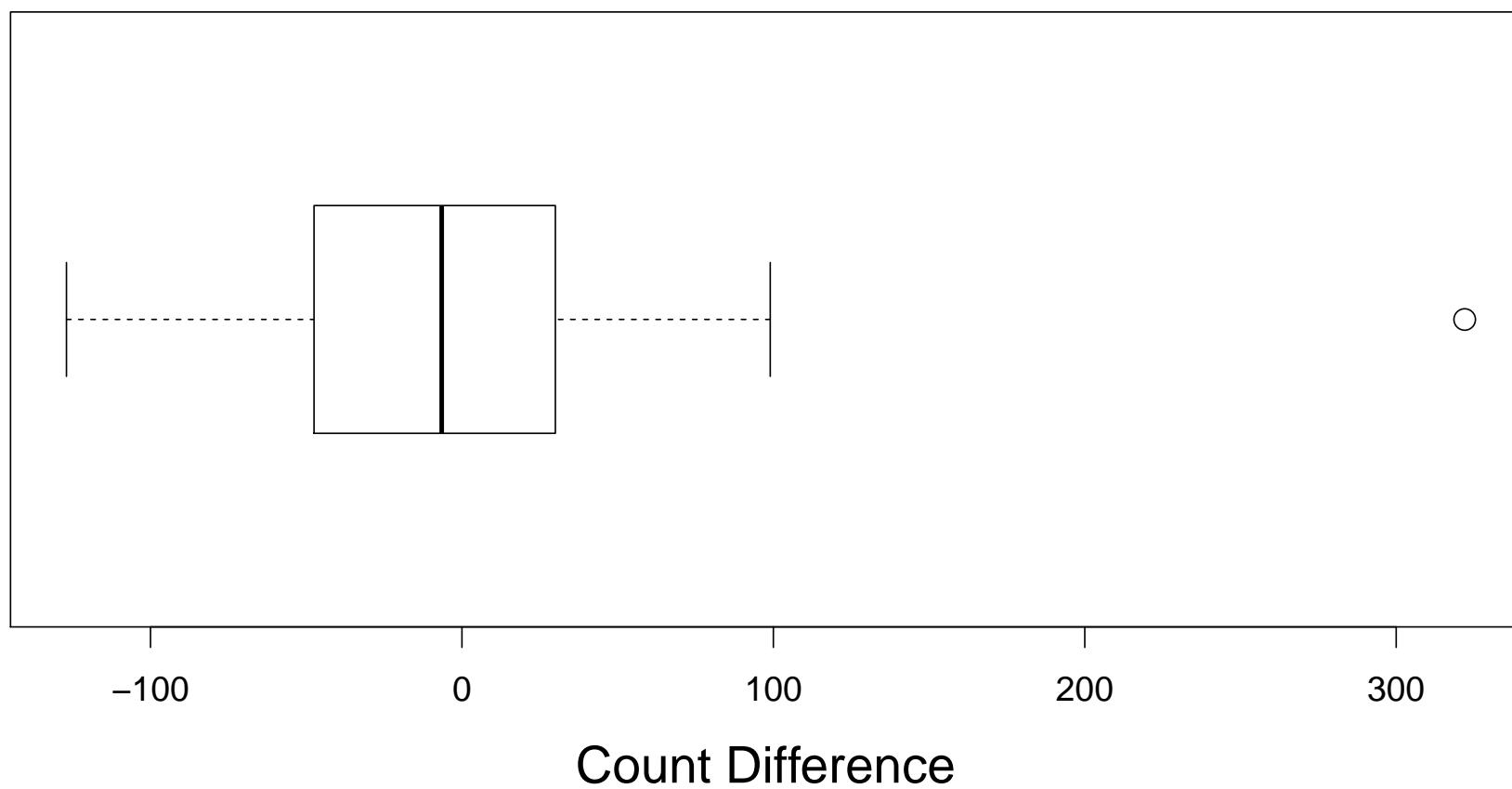
What is the research question?

Consider the following dataset of bird counts in 1992 and 2012. How to graph the data depends what we want to know.

1992:	47	13	13	0	46	222	104	35	110	969	74	0
2012:	0	61	0	0	58	95	0	0	432	1068	26	0

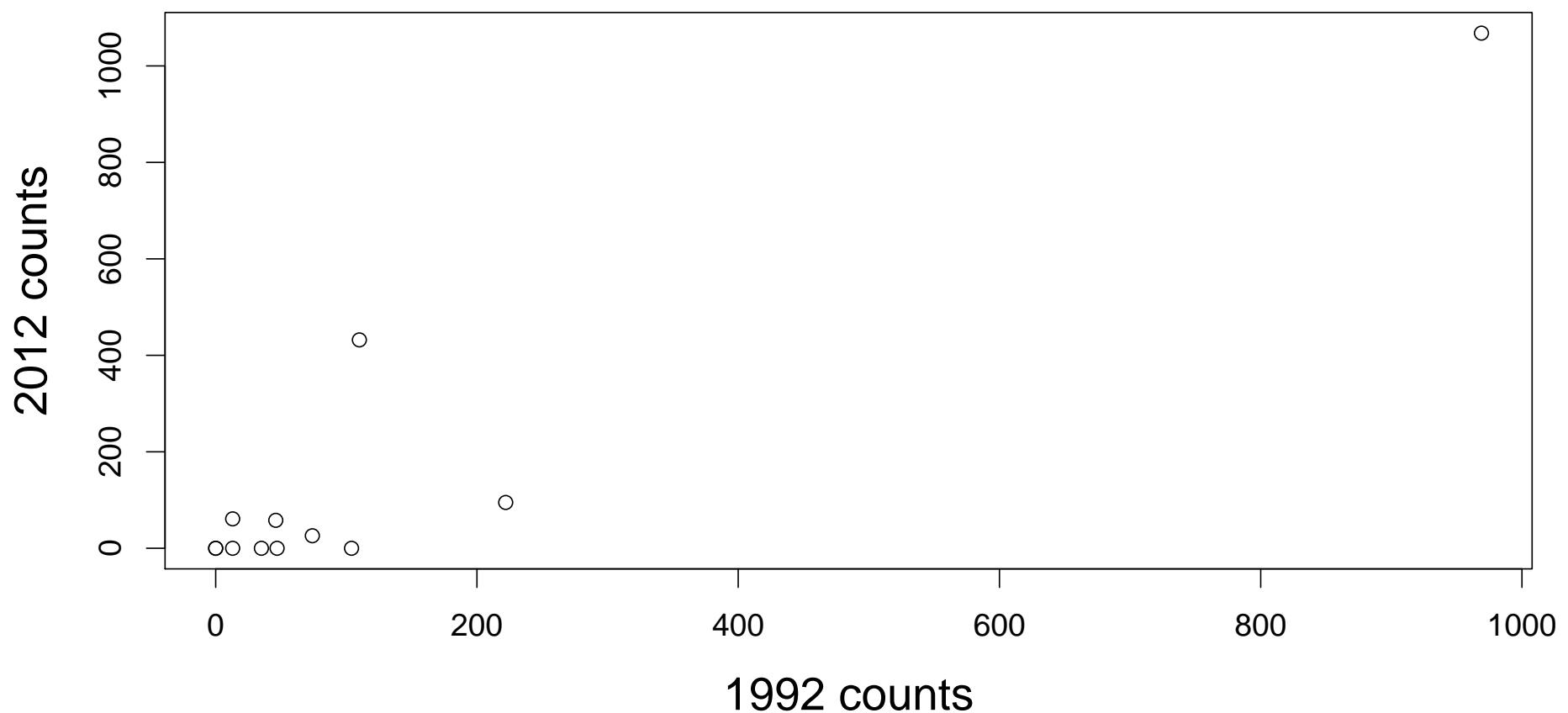
Graph 1 – are counts larger in 2012 than in 1992?

Boxplot of differences in counts between 2012 and 1992



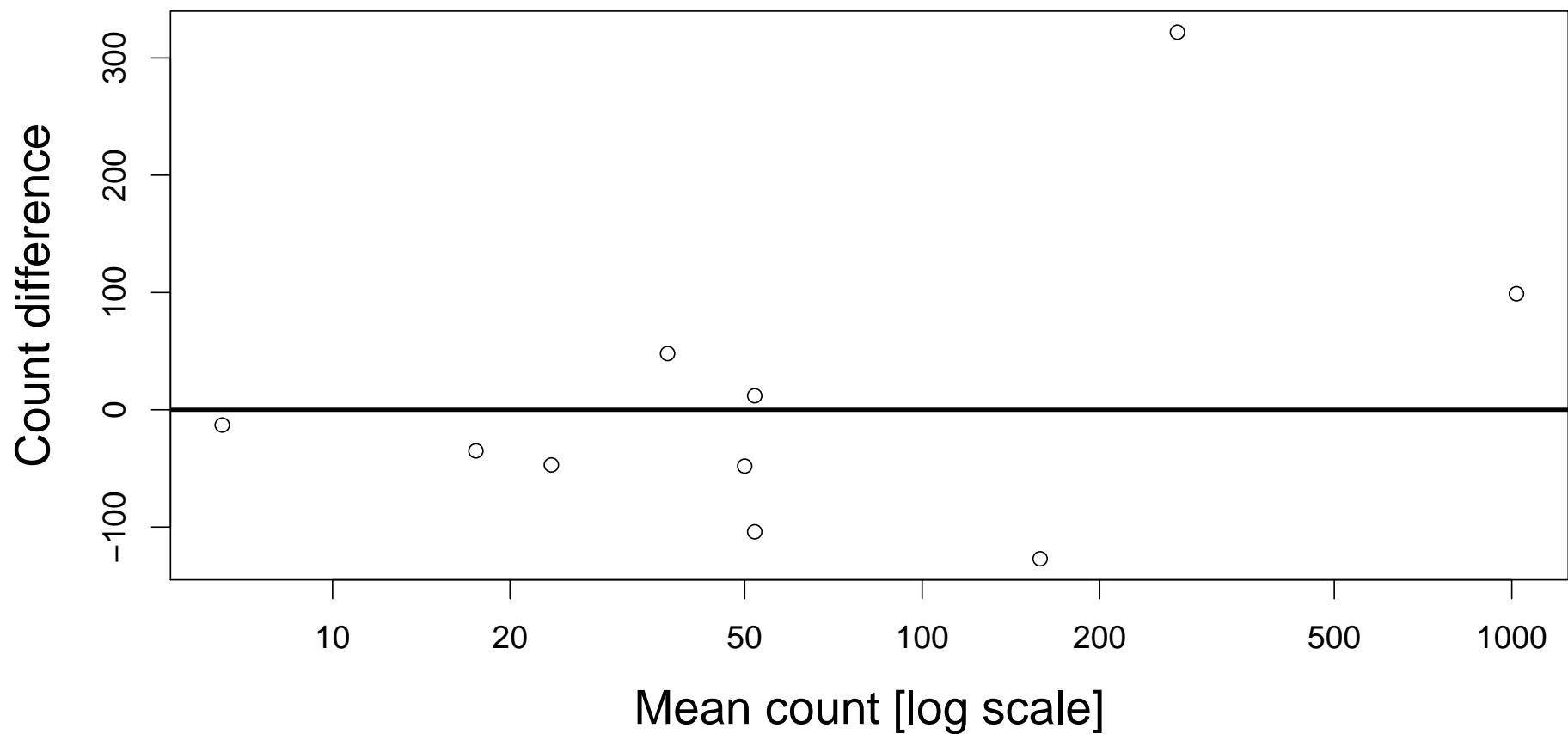
Graph 2 – Is 2012 count related to 1992 count?

Scatterplot of 2012 counts vs. 1992 counts



Graph 3 – Do counts for different years measure the same thing?

Difference in counts vs. mean counts (Mean–Difference Plot)



What is the research question?

1992:	47	13	13	0	46	222	104	35	110	969	74	0
2012:	0	61	0	0	58	95	0	0	432	1068	26	0

The right approach to analysis depends on the question!

Main types of approaches to analysis

- Descriptive statistics (exploring the data)
- Hypothesis testing (*a priori* hypothesis of key interest)
- Estimation/confidence intervals (to estimate the key quantity/effect size)
- Predictive modelling (to predict some key response variable)
- Variable selection (finding which variables are related to response)

Descriptive or inferential statistics?

Descriptive if the goal is to **describe** – explore and understand the main features of data.

- Graphs: histograms, scatterplots, barplots, ...
- Numbers: mean, median, sd, ...
- “Pattern-finders” /data-mining tools: cluster analysis, ordination

Descriptive or inferential statistics?

Inferential if the goal is to **infer** – making general statements beyond the data at hand.

- Hypothesis tests: t -tests, binomial tests, χ^2 test of independence, ANOVA F -test, ...
- Confidence intervals: for mean difference, regression slope, some other measure of effect size
- Predictive modelling and variable selection

Inferential statistics requires assumptions about data and study design
– **always check assumptions before making inferences!**

Which method do you use when?

When thinking about how to analyse data, there are two key things to think about:

- What is the main research question?
- **What are the main properties of my data?**

What are the main properties of my data?

Of particular interest:

- Categorical or quantitative?
- For quantitative: discrete (esp. if lots of small counts near zero) or continuous?
- One variable, association between two variables, three, ...
- Design considerations: any pairing (or blocking)?

In regression problems (*i.e.* everything over the next two days), what really matters are the properties of the **response variable** (the one you are trying to predict)

Data types – categorical or quantitative?

This is a key distinction.

categorical – breaks subjects into categories (e.g. colour, species ID)

quantitative – measured on a scale (e.g. biomass, species richness)

Data types – types of categorical

Two types of categorical data that sometimes pop up:

ordinal – categories that have a natural ordering (e.g. abundance as one of {absent, present in trace amounts, a fair bit, bloody everywhere})

nominal – not ordinal (e.g. colour as red, blue, green, purple, ...)

(We won't worry about this stuff though.)

Data types – types of quantitative

For quantitative variables, sometimes we distinguish between discrete and continuous data:

discrete – quantitative data that takes a “countable” number of values, like 0, 1, 2, ... (e.g. species richness)

continuous – quantitative data that can take any value in some interval (e.g. biomass, plant height)

Discrete vs continuous is an important distinction for “highly discrete” with lots of zeros. Larger counts you can usually treat as continuous.

Example: data types

Which type of data is each of the following variables – categorical or quantitative? Discrete or continuous? Ordinal or nominal?

- Gender
- Bird count
- Sampling time (one of 1992 and 2012)

Data analysis for one or two variables

variable type:	one variable		two variables		
	categorical	quantitative	both categorical	one categorical, one quantitative	both quantitative Paired data?
useful graphs:	bar chart	boxplot or histogram	clustered bar chart	comparative boxplots	scatterplot
useful numbers:	table of frequencies	mean and sd or 5-number summ.	2-way table of frequencies	5-num for each group	correlation or regression
useful test:	1-samp for p (for binary var)	1-sample t -test (or Z-test if σ known)	χ^2 test for independence	2-sample t (for binary+quant)	test regression slope (β_1)
useful for inference:	CI for p (for binary var)	CI for μ		CI for $\mu_1 - \mu_2$	CI for β_1
	(analyse differences)		Paired data?		

Example: gender bias

Kerry goes bat counting. She finds 65 female bats and 44 male bats in a colony. She would like to know if there is evidence of gender bias.

What do the data tell us – one variable or two? Categorical or quantitative?

What does the question tell us – descriptive, estimation, hypothesis testing, etc?

What graph would you use to visualise the data?

So how would you analyse the data?

Example: Pregnancy and smoking

(Journal of General Psychology 1993, 120(1): 49-63)

What is the effect of a mother's smoking during pregnancy on the resulting offspring?

A randomised controlled experiment investigated this by injecting ten pregnant guinea pigs with 0.5 mg/kg nicotine hydrogen tartrate in saline solution. The ten guinea pigs in the control group were injected with a saline solution without nicotine.

The learning capabilities of the **offspring** of these guinea pigs was then measured using the number of errors that were made trying to find food in a maze.

The number of errors made by guinea pigs in the maze:

Control	11	19	15	47	35	10	26	15	36	20
Treatment	38	26	33	89	66	23	28	63	43	34

What do the data tell us – one variable or two? Categorical or quantitative?

What does the question tell us – descriptive, estimation, hypothesis testing, etc?

What graph would you use to visualise the data?

So how would you analyse the data?

Statistical inference

Inference about what?

About general (“population”) patterns, based on your data (“sample”).

Notation: sample estimates use the regular alphabet, population parameters usually use Greek letters (e.g. s vs σ). But sometimes sample estimates have hats on them (e.g. $\hat{\beta}$ vs β).

	Sample	Population
Mean	\bar{x} (or \bar{y})	μ (or μ_y)
SD	s	σ
Proportion	\hat{p}	p
Regression slope	$\hat{\beta}$	β

We usually refer to the number of observations in our sample as the “sample size”, n .

Inference notation – bat example

Kerry goes bat counting. She finds 65 female bats and 44 male bats in a colony. She would like to know if there is evidence of gender bias.

We have a sample of $n = 65 + 44 = 109$ bats.

The estimated proportion of female bats is $\hat{p} = \frac{65}{65+44} \simeq 0.596$.

We are interested in the true proportion of bats in the colony that are female, p , which is unknown.

Inference notation – bird example

e.g. 1992 bird counts: 47, 13, 13, 0, 46, ..., sample mean is $\bar{x}_{1992} = 152$.

e.g. 2012 bird counts: 0, 61, 0, 0, 58, ..., sample mean is $\bar{x}_{2012} = 46$.

But we are interested in whether there has been a change in the true mean number of birds – if the true means are μ_{1992} and μ_{2012} respectively, we want to make inferences about:

$$\mu_{1992} - \mu_{2012}$$

Two common types of inference

There are two main ways to make inferences about the true value of some parameter, e.g. p , the sex ratio in the bat colony.

Confidence interval – we don't know the true sex ratio, but we can construct an interval which we are pretty sure contains the true sex ratio.

e.g. we are 95% confident that the true p is between 0.498 and 0.688.

Hypothesis test – the null hypothesis of no gender bias is $H_0 : p = 0.5$. We can use probability to work out how likely it is to get as many as 65 female bats in a random sample of 109 if there is no gender bias:

$$P\text{-value} = 2 \times P\left(\hat{p} > \frac{65}{109}\right) = 0.055$$

This is reasonably unlikely (almost 5%) so there is reasonable evidence against H_0 .

Interpreting P -values

A P -value measures how unlikely your data are, under H_0 . This can be used as a measure of how much evidence there is **against H_0** (but **not** as a measure of evidence for H_0).

Small P -value \Rightarrow data unlikely under $H_0 \Rightarrow$ evidence against H_0 .

Large P -value \Rightarrow data not unlikely under $H_0 \Rightarrow$ no evidence against H_0 .

Often easier to interpret if you have a confidence interval as well.

Scenario 1 – evidence against H_0

75 females out of 109 bats $\Rightarrow p\text{-value} = 0.001 \Rightarrow$ strong evidence (against H_0) of gender bias.

We are 95% confident that the true sex ratio is in the interval (0.59, 0.77) – so you can see there is clear evidence that the proportion of females is bigger than 0.5.

Scenario 2 – no evidence against H_0

57 females out of 109 bats $\Rightarrow p\text{-value} = 0.70 \Rightarrow$ no evidence (against H_0) of gender bias.

We are 95% confident that the true sex ratio is in the interval (0.43, 0.62). 0.5 is in the interval so we don't have evidence of a bias towards females. But we can't rule it out, maybe the true proportion is 0.6!

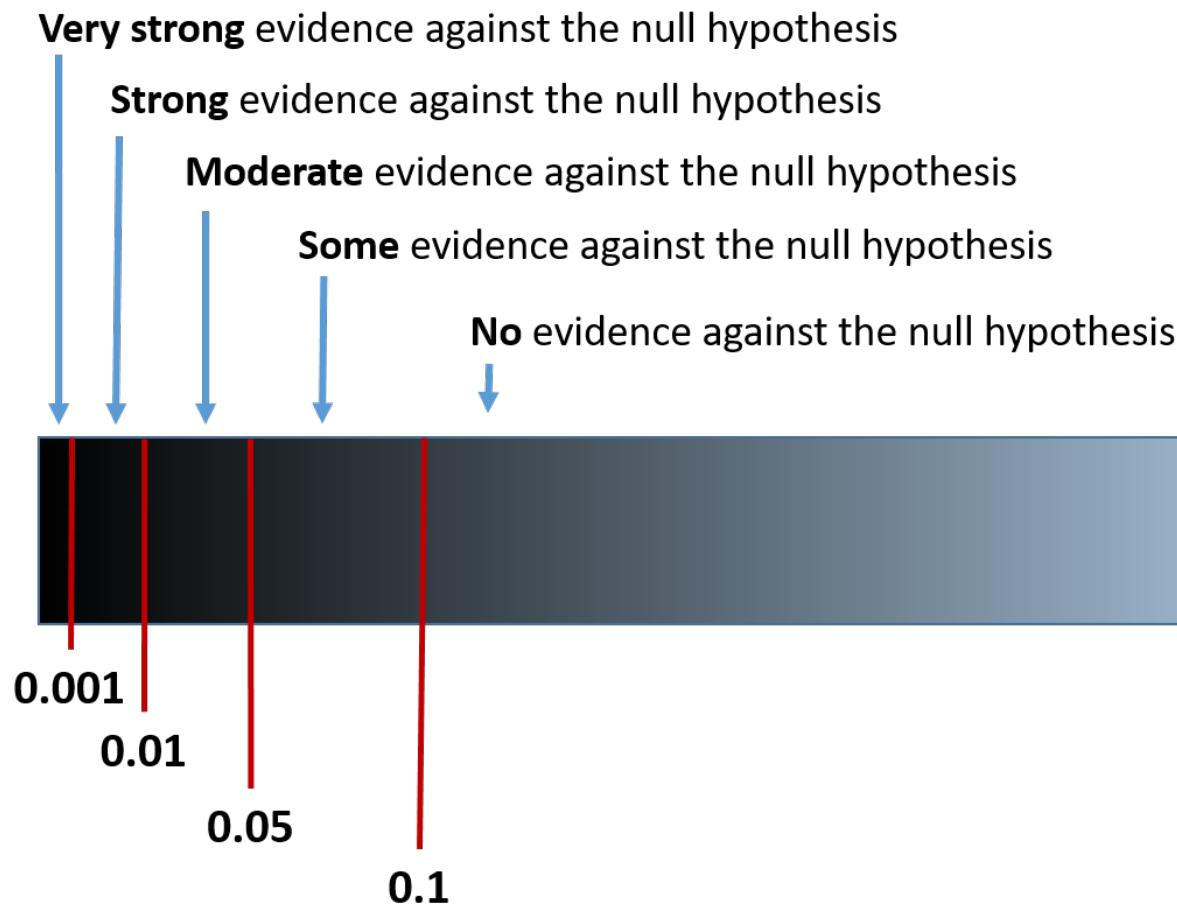
So a large P -value doesn't mean that H_0 is true. Always helps to look at the confidence interval to interpret.

What's the cut-off?

<u>P-VALUE</u>	<u>INTERPRETATION</u>
0.001	
0.01	
0.02	HIGHLY SIGNIFICANT
0.03	
0.04	
0.049	SIGNIFICANT
0.050	OH CRAP. REDO CALCULATIONS.
0.051	ON THE EDGE
0.06	OF SIGNIFICANCE
0.07	HIGHLY SUGGESTIVE,
0.08	SIGNIFICANT AT THE
0.09	P<0.10 LEVEL
0.099	HEY, LOOK AT
≥0.1	THIS INTERESTING SUBGROUP ANALYSIS

What's the cut-off?

You will undoubtedly hear people and papers declare things to be **statistically significant** if the p-value for the test is less than **0.05**. Most statisticians find this cut-off arbitrary and frustrating, but nevertheless some kind of cut off seems to be necessary. It's often more useful to talk about **evidence** rather than significance, because at least that way it's clear that you are not absolutely certain of the outcome just because your p-value falls below an arbitrary cut-off.



Null hypotheses (H_0)

The null hypothesis is generally what is assumed to be true until evidence indicates otherwise. Most commonly it is either: that there are no differences between the means of groups; no association or relationship between variables (or in the bat example it is that the proportion of males (females) is 50%). We conduct experiments to find evidence against the null hypothesis (no difference/association) and in support of an alternative hypothesis (a difference/association).

Use and abuse of hypothesis tests

Hypothesis tests are often abused or misused! Main misuses:

- Concluding that H_0 is true because P is large
This one can be avoided by also looking at your question in terms of estimation (with confidence intervals) – what is a range of plausible values for the effect size
- Testing hypotheses you didn't collect the data to test
- "Searching for significance" with lots of hypothesis tests
- Testing claims you know aren't true in the first place.

Confidence intervals

Most sample estimates of parameters ($\bar{x}, \hat{p}, \hat{\beta}, \dots$) are approximately normally distributed, and most software will report a standard error (standard deviation of the estimator) as well as its estimate. In such cases an approximate 95% confidence interval for the true (“population”) value of the parameter (μ, p, β, \dots) is:

$$(\text{estimate} - 2 \times \text{standard error}, \text{estimate} + 2 \times \text{standard error})$$

About 95% of the time, such an interval will capture the true value of the parameter, if assumptions are satisfied.

You can often compute confidence intervals on R using the `confint` function.

Assumptions

When computing a confidence interval or a P -value, there are always assumptions. You need to check they are reasonable!

Assumptions of a one-sample test of proportions (for bat gender):
Each observed bat in the sample has the same probability of being female, independent of (*i.e.* unaffected by) the gender of any other bats in the sample (“identically and independently distributed”, “iid”).

This assumption is **guaranteed by random sampling** – if the bats in the sample are a random sample from those in the colony then it will be satisfied.

Nearly all statistical models/tests/CIs involve an independence assumption, and it can be guaranteed by randomly sampling or in experiments by randomising allocation of subjects to treatments.

Two-sample t test

Is number of errors made by guinea pigs related to nicotine treatment?

Control	11	19	15	47	35	10	26	15	36	20
Nicotine	38	26	33	89	66	23	28	63	43	34

We can answer this question using a **two-sample t -test**

Two-sample t -test as a test of association

Rather than thinking of this as comparing # errors in two samples (control and nicotine) think of it as:

Is there an association between # errors and treatment group?

That is, we think of this problem as having two variables – one quantitative and one categorical.

So the two-sample t -test is **a test for association** between # errors and treatment.

The two-sample t -test

We test H_0 : no association between # errors and treatment using:

$$t = \frac{\bar{y}_{\text{Nicotine}} - \bar{y}_{\text{Control}}}{\text{standard error of } \bar{y}_{\text{Nicotine}} - \bar{y}_{\text{Control}}}$$

which (if H_0 is true) comes from a t distribution with degrees of freedom $n - 2$ where n is the total sample size ($n = n_{\text{Nicotine}} + n_{\text{Control}}$).

Confidence intervals are returned automatically when performing the test on R, but they are calculated in the usual way.

```
> datasmoke <- read.csv("smokePregnant.csv")
> t.test(datasmoke$Nicotine,datasmoke$Control, var.equal=TRUE)
```

Two Sample t-test

```
data: datasmoke$Nicotine and datasmoke$Control
t = 2.671, df = 18, p-value = 0.01558
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 4.460667 37.339333
sample estimates:
mean of x mean of y
        44.3       23.4
```

t-test assumptions

In a two-sample *t* test of data y we assume that:

1. The y -values are **independent** in each sample, and the two samples are independent
2. The y -values are **normally distributed** with **constant variance**

$$y_{ij} \sim \mathcal{N}(\mu_i, \sigma^2)$$

Checking *t*-test assumptions

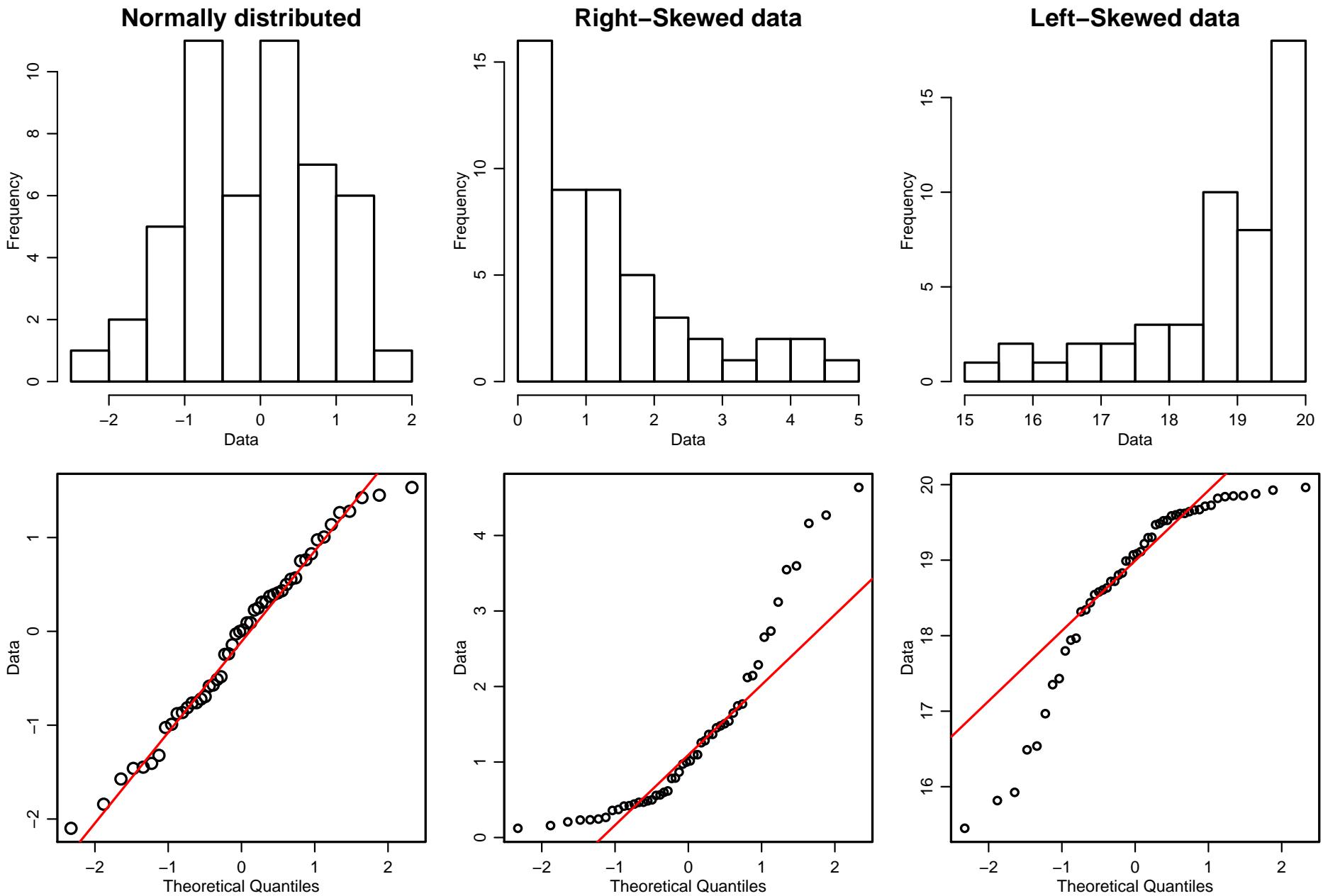
1. **Independent** y can be guaranteed – **how???**
2. **normality** can be checked with a **normal quantile plot**
Constant variance can be checked by comparing standard deviations. Or using a residual vs fits plot (as is done later for linear regression).
Don't worry about different sd's if sample sizes are the same (but consider transformations, as we discuss shortly).

Normal quantile plot of residuals

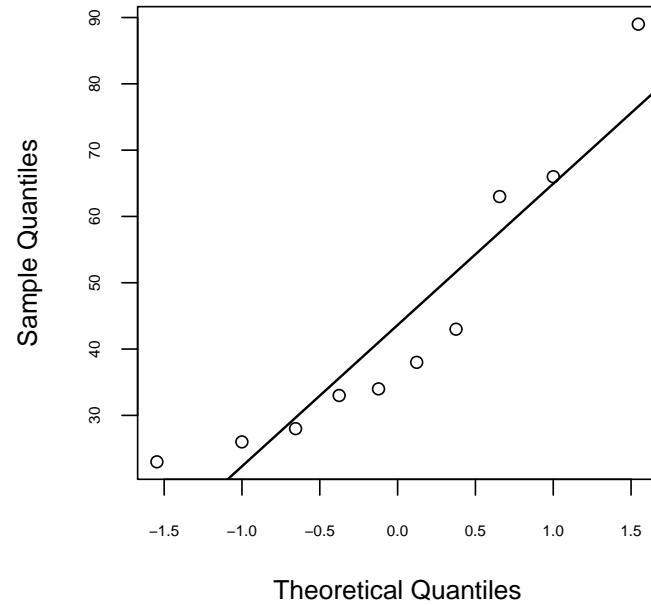
Normal quantile plots are the best method of checking if a variable is normally distributed – they plot the variable against values expected from a normal distribution.

Don't be overly worried – this assumption doesn't matter unless:

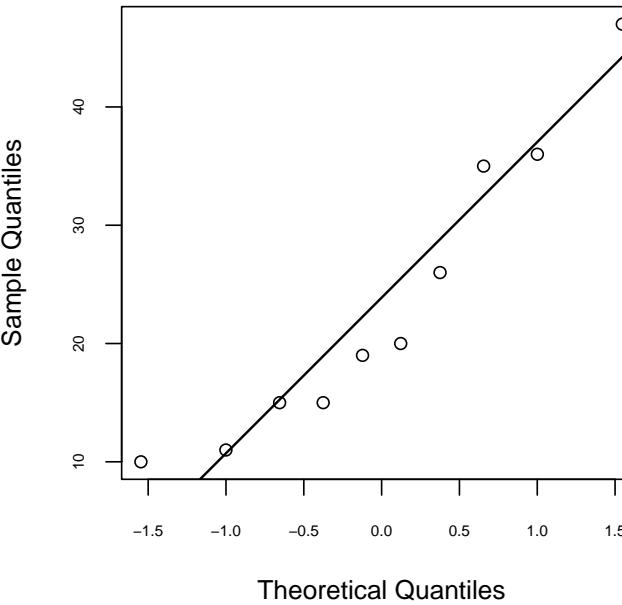
- you have a small sample size (e.g. $n < 10$)
- Data are strongly skewed or have big outliers



Normal quantile plot of Nicotine observations



Normal quantile plot of Control observations



```
> qqnorm(datasmoke$Nicotine); qqline(datasmoke$Nicotine)
> qqnorm(datasmoke$Control); qqline(datasmoke$Control)
> sd(datasmoke$Nicotine)
[1] 21.46858
> sd(datasmoke$Control)
[1] 12.30357
```

Do you think assumptions are reasonable?

Hypothesis tests of assumptions suck

There are plenty of formal tests around for checking assumptions:

- Tests of normality: Anderson-Darling, Shapiro-Wilk, ...
- Levene's test, F -test, ...

These should generally be avoided. Checking a graph is fine.

(Remember you should only test hypotheses you collected the data to test!)

Transformations

Consider y_{new} , formed as some function of a variable y . Examples:

$$y_{\text{new}} = y^2 \quad y_{\text{new}} = 32 + \frac{9}{5}y \quad y_{\text{new}} = \log(y)$$

If y_{new} is a function of y , we say y_{new} is a transformation of y . The act of calculating y_{new} is referred to as **transforming** y .

Why transform data? **To change its shape**, in particular, to get rid of strong skew and outliers. (t -tests and also regression don't work well for such data, particularly for small sample sizes)

Note that linear transformations don't change the shape, they only change the scale. Only non-linear transformations are shape-changers.

How transformation changes the shape of data.

Common transformations

When your data takes positive values ($y > 0$) and is right-skewed, these transformations might make it more symmetric:

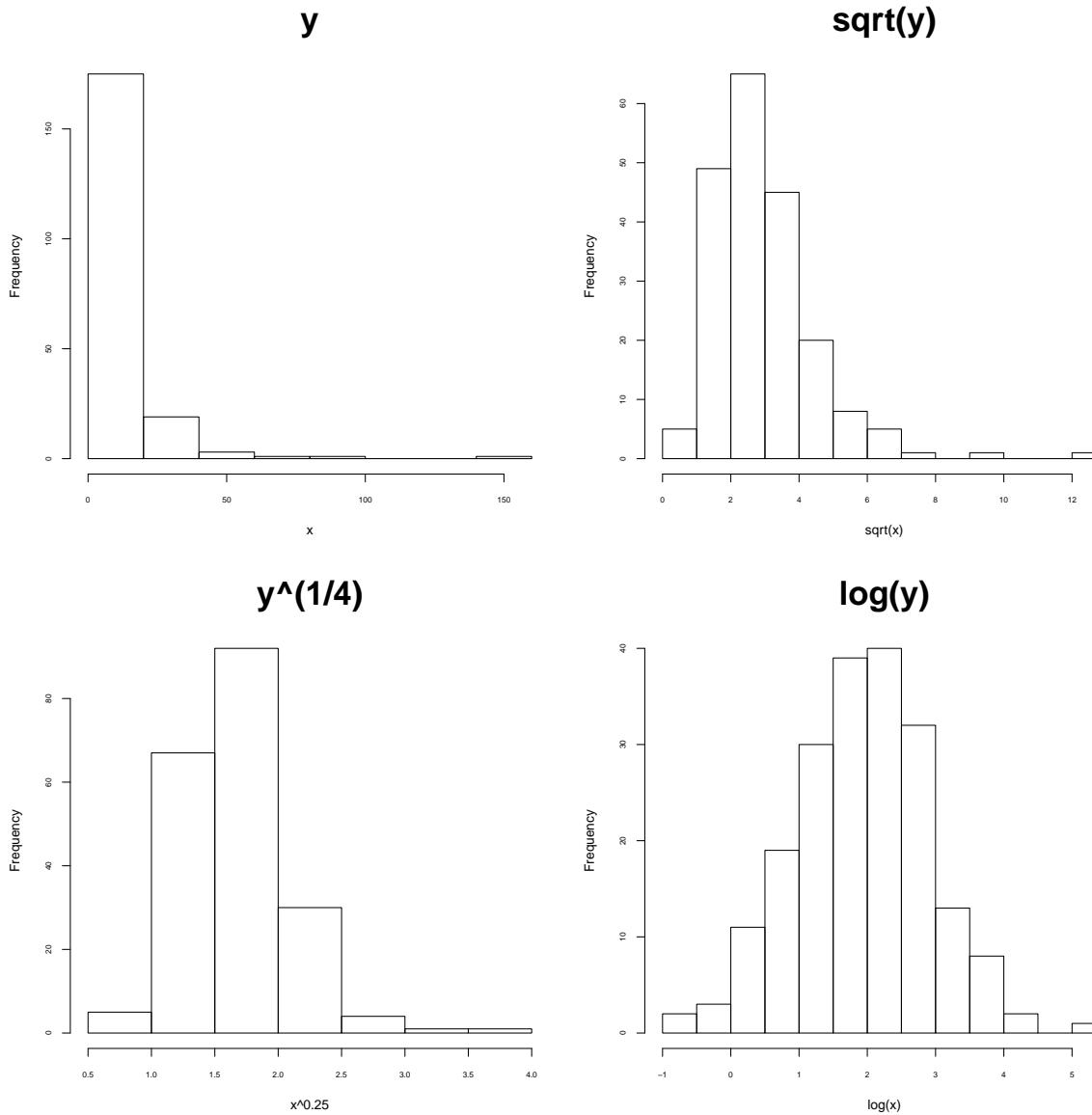
- $y_{\text{new}} = \sqrt{y}$
- $y_{\text{new}} = y^{1/4}$
- $y_{\text{new}} = \log y$

These transformations are all “concave down”, hence they reduce the length of the right tail. They are in increasing order of strength – that is, for strongly skewed data, is more likely $y_{\text{new}} = \log y$ to work than $y_{\text{new}} = \sqrt{y}$.

They are also monotonically increasing – that is, as y gets larger, y_{new} gets larger.

(A transformation which didn't have this property would be hard to interpret!)

Examples:



Log transformation

The logarithmic or log-transformation is particularly important:

$$y_{\text{new}} = \log_a y$$

where a is the “base”, commonly $\log_{10} y$, $\log_e y$, $\log_2 y$.

(The base doesn’t matter – it only affects the scale, not the shape.)

Logs have the following key property:

$$\log(ab) = \log a + \log b$$

and more generally,

$$\log(y_1 \times y_2 \times \dots \times y_n) = \log y_1 + \log y_2 + \dots + \log y_n$$

In words, **the logarithm transforms multiplicative to additive.**

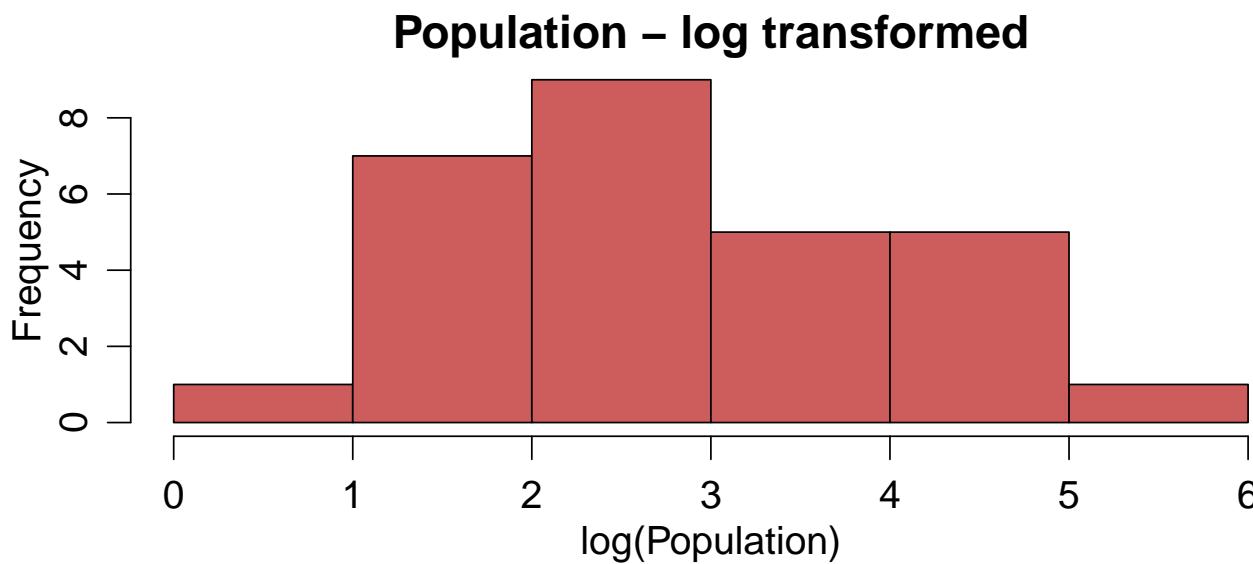
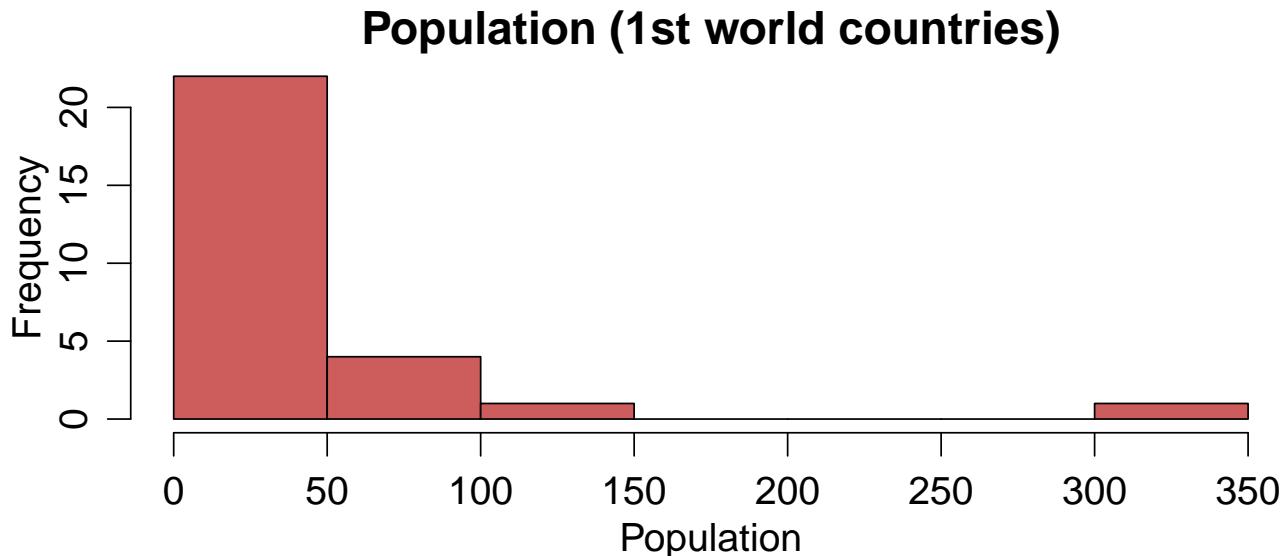
Many variables can often be understood as the outcome of a series of multiplicative processes.

Examples:

- Wealth
- Size
- Profit
- Population

By transforming such processes, they change from multiplicative to additive.

Example:



Another situation where you need a transformation – boundaries

If your data are “pushed up” against a boundary, you may think about transforming data to remove the boundary.

e.g. population can be small but it can't be negative, so values are pushed up against zero. A log-transformation removes this boundary (because as y approaches 0, $\log(y)$ approaches $-\infty$).

Boundaries are a problem in regression – can lead to nonsensical predictions, e.g. a predicted population of -2 million!

Situations needing a different transformation

Proportions: if data are between 0 and 1, try the logit transformation:

$$y_{\text{new}} = \log \left(\frac{y}{1 - y} \right)$$

(This stretches data over the whole real line, from $-\infty$ to ∞ .)

The arcsine transform was often used historically – not a good idea.

Right-skewed with zeros: This often happens when data are counts. The problem is that you can't take logs because $\log 0$ is undefined. Try:

$$y_{\text{new}} = \log(y + 1)$$

Left-skewed data: This is less common. But if data are left-skewed and negative, then $-y$ is right-skewed and positive, in which case the transformations previously discussed can be applied to $-y$.

e.g. $y_{\text{new}} = \log(-y)$ takes negative, left-skewed values of y and tries to make them more symmetric.

Linear Regression

- Simple Linear Regression
- Equivalence of two-sample t -test and linear regression

Simple linear regression

Linear regression when there is only **one** x variable.

Example: Water quality

How is water quality in creeks associated with size of the catchment area? Water catchment area and an index measuring water quality (IBI) are calculated for a sample of 20 creeks:

Catchment area (km ²)	29	49	28	8	57	...	26
Water quality (IBI)	61	85	46	53	55	...	85

What does the research question ask us – descriptive, estimation, hypothesis testing, etc?

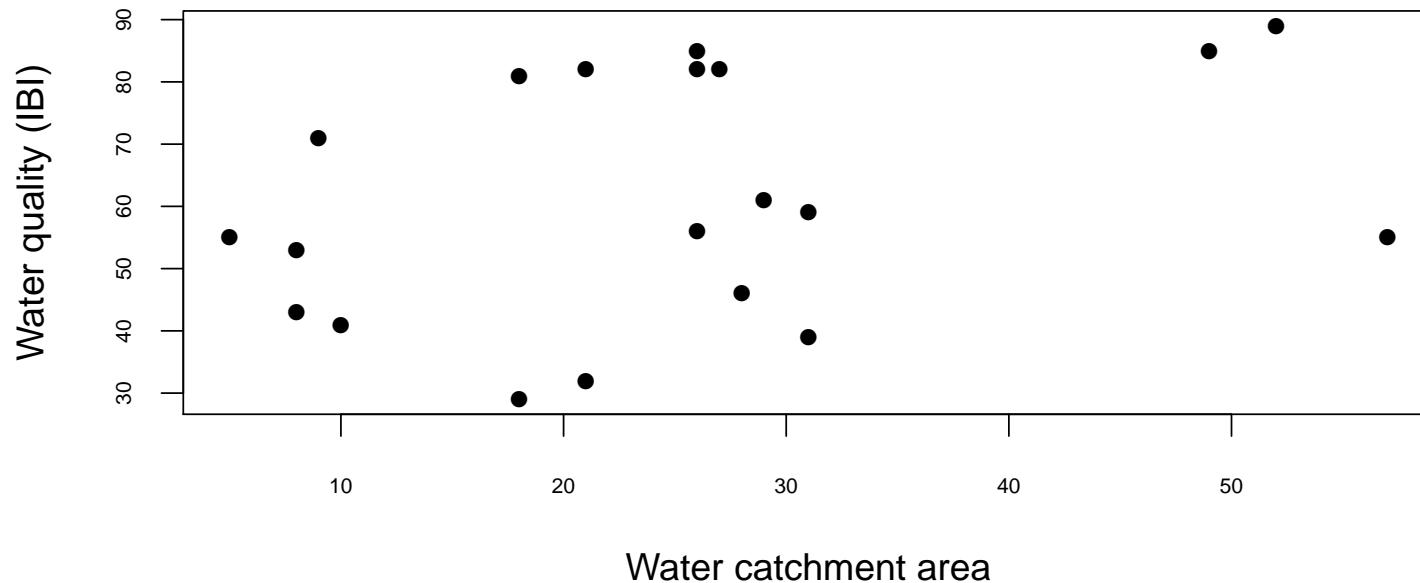
What do the data tell us i.e., its properties – one variable or two?
Categorical or quantitative?

What graph would you use to visualise the data?

So how would you analyse the data?

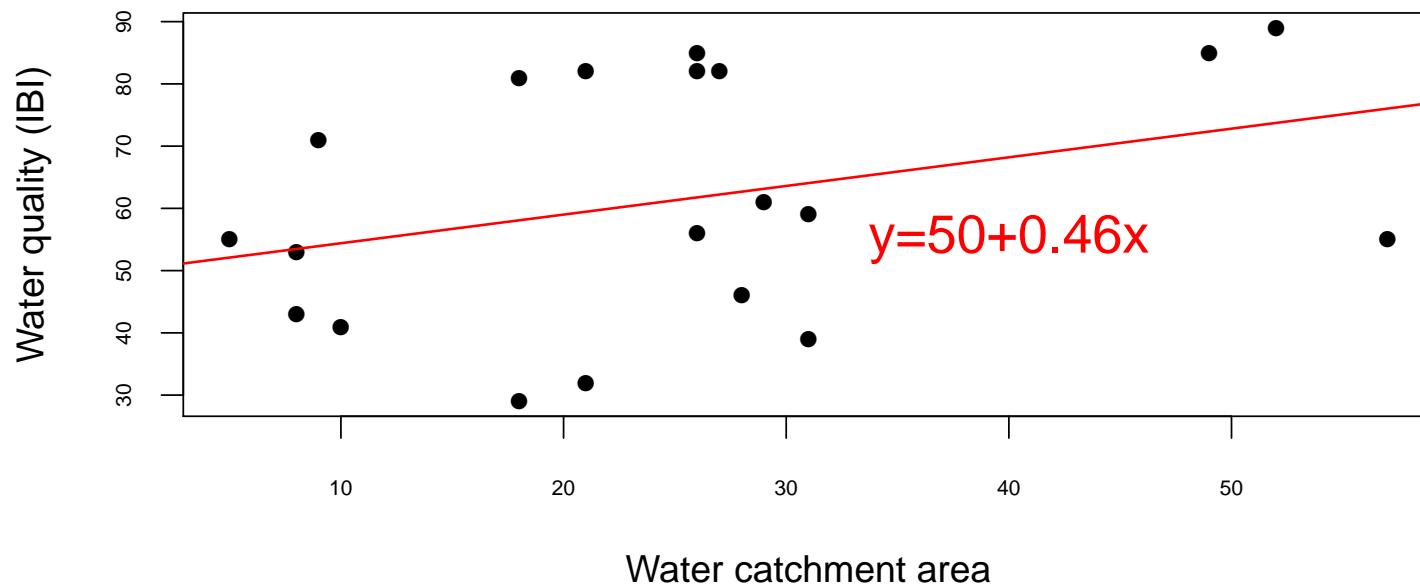
Water quality vs catchment area: scatterplot

Scatterplot of water quality vs. catchment area

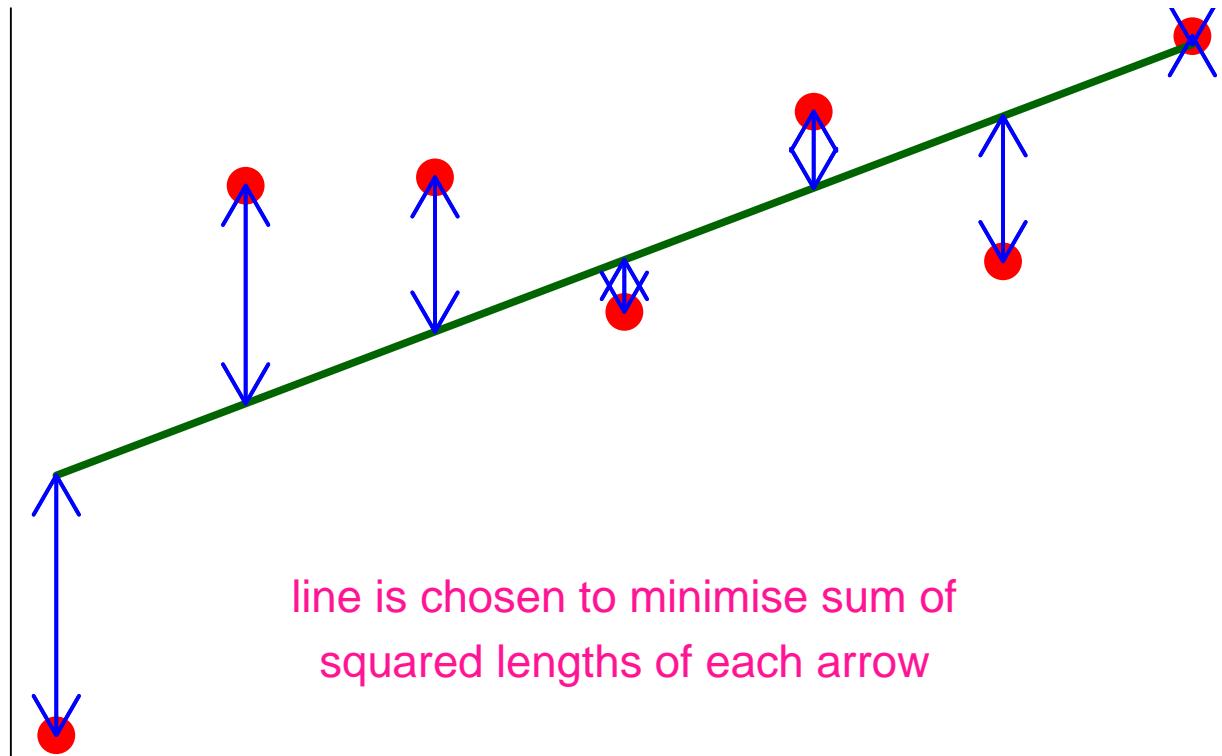


Water quality vs catchment area: linear regression

Scatterplot of water quality vs. catchment area



How is a regression line fitted? **Least squares**



<http://hspm.sph.sc.edu/COURSES/J716/demos/LeastSquares/LeastSquaresDemo.html>

Mathematical Representation of regression Line

Simple linear regression is a straight line, taking the algebraic form

$$\mu_y = \beta_0 + \beta_1 x,$$

where

β_0 = intercept on y-axis ($x=0$)

β_1 = slope of the line.

β_1 represents the **magnitude of the effect of x on y**.

Useful diagram:

The water quality example: linear regression in R

```
> datqual <- read.csv(file = "waterQual.csv")
> fit.qual1=lm(quality~catchment, data = datqual)
> summary(fit.qual1)
```

Call:

```
lm(formula = quality ~ catchment)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	49.7945	8.5246	5.841	1.56e-05 ***
catchment	0.4602	0.2966	1.552	0.138

Residual standard error: 18.81 on 18 degrees of freedom

Multiple R-squared: 0.118, Adjusted R-squared: 0.06898

F-statistic: 2.408 on 1 and 18 DF, p-value: 0.1381

The water quality example: linear regression in R

The model is:

$$\mu_{\text{quality}} = \beta_0 + \text{catchment} \times \beta_1$$

Find an approximate 95% confidence interval for β_1 .

Explain what β_1 means in words.

N.B. If you want you can get more accurate confidence intervals in R, use `confint`.

The importance of testing slope= 0

Notice there is a t -statistic and P -value for each coefficient. These test the null hypothesis that the true value of the coefficient is 0.

Why 0?

For y -intercept – no reason, usually, it's irrelevant!

For slope – this tests for an **association between y and x .**

Note the use of the word “association”, as for our two-sample t -test

Useful diagram:

Assumptions of linear regression

For inference about simple linear regression, we assume:

1. The observed y values are **independent**, after accounting for x .
2. The y values are **normally distributed with constant variance**.

$$y_i \sim \mathcal{N}(\mu_i, \sigma^2)$$

3. There is a **straight line relationship** between mean of y and x

$$\mu_i = \beta_0 + \beta_1 x_i$$

N.B. Note the first two assumptions are almost the same as for the two-sample t -test

Assumptions of linear regression

1. **Independence of y** can be guaranteed to be satisfied – **how???**
2. **Normality** doesn't really matter (due to Central Limit Theorem) except for small samples/strongly skewed data/outliers. Check on a **normal quantile plot** of residuals.
Constant variance can be checked using a **residuals vs fits plot** too see if there is any fan-shape pattern
3. **A straight line relationship** is crucial – no point fitting a straight line to non-linear data! Check for no obvious pattern e.g., a U-shape, on a **residual vs fits plot**.

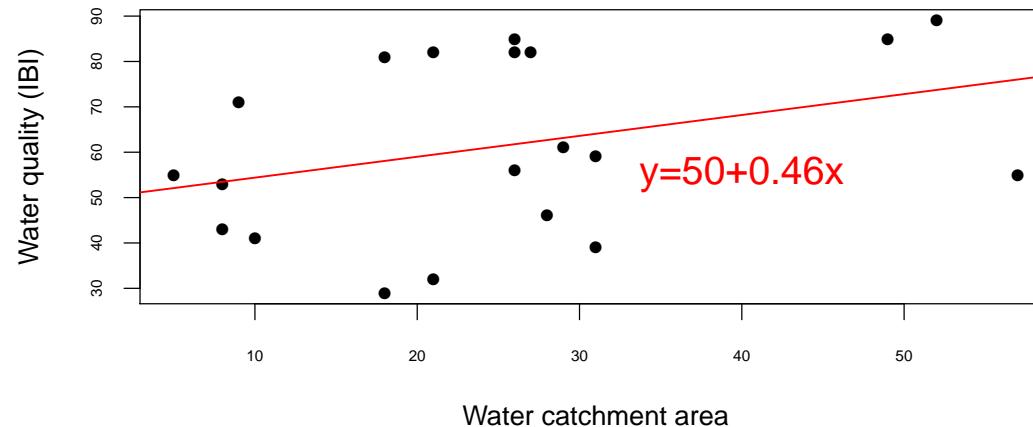
Both a normal quantile plot of residuals and a residual vs fits plot can be produced using `plot(fit.qual1)`

Residual vs fits plot

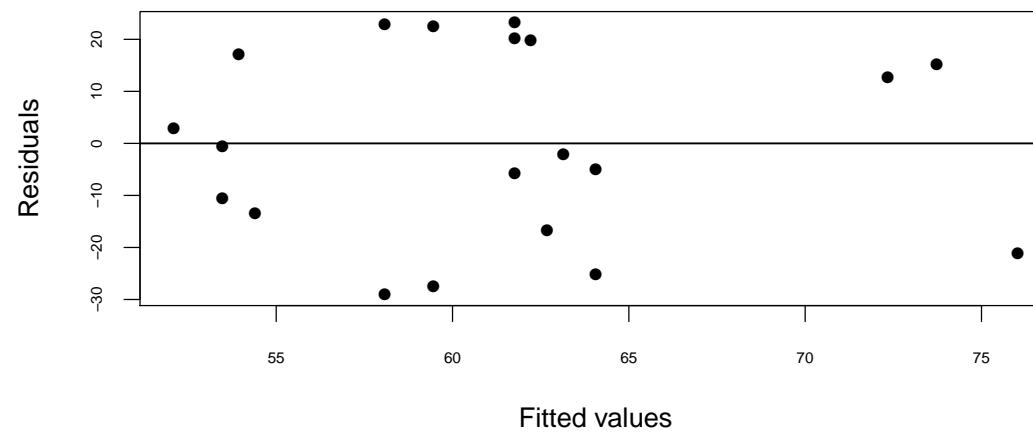
A **residual vs fits plot** ($y - \hat{\mu}_y$ vs $\hat{\mu}_y$) can be used to check if the data are linearly related and if the variance is constant for different values of x .

There should be **no pattern** on a residual plot – if there is, then one or more of the assumptions made above are violated, and we can **not** make inferences about the true regression line using simple linear regression.

Scatterplot of water quality vs. catchment area

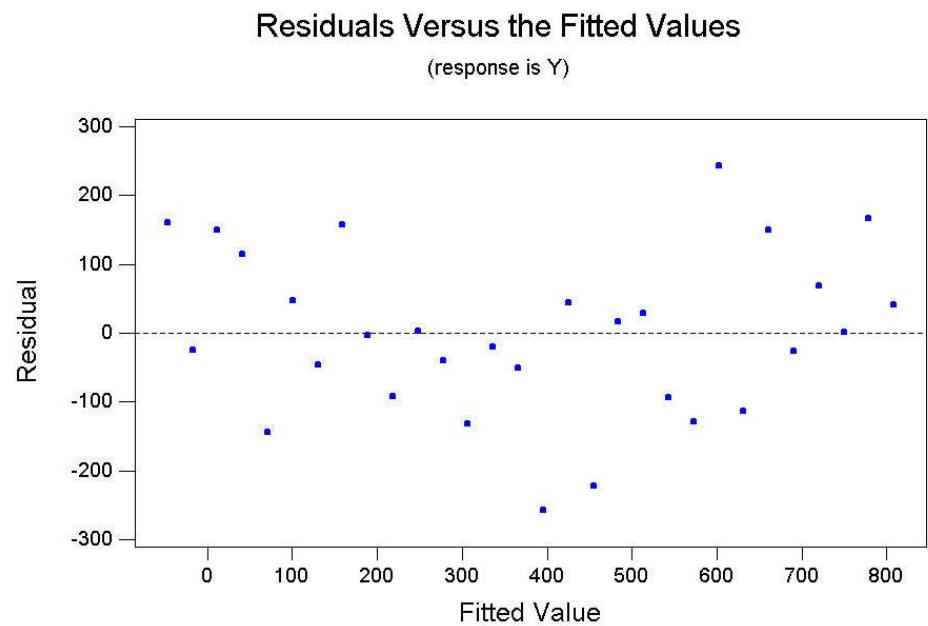
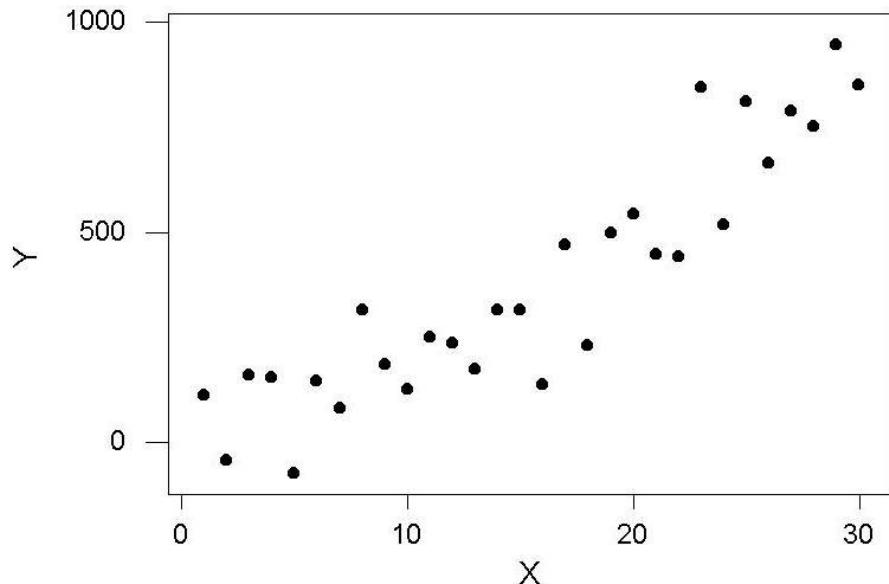


Residual vs. fits plot for water quality data



Scary patterns in residual vs fits plots

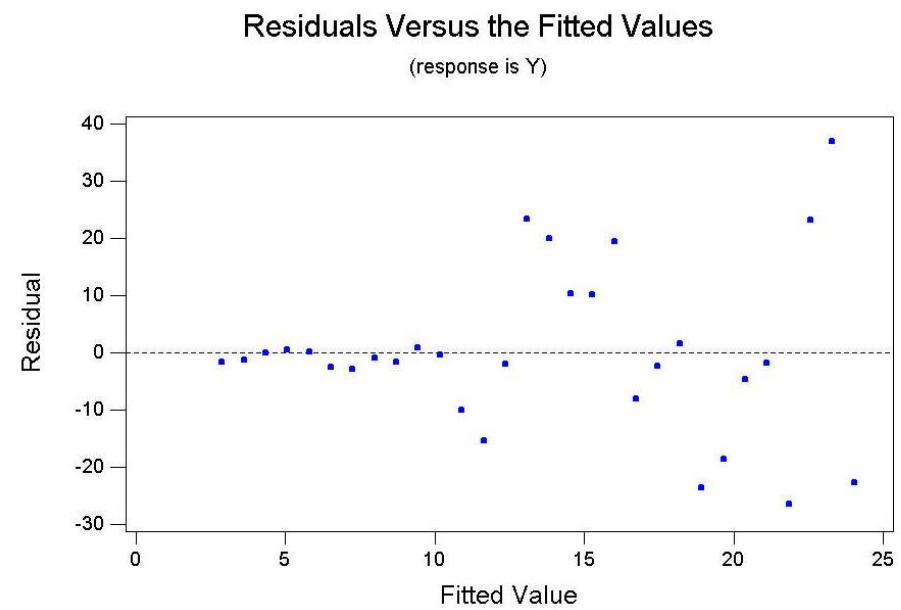
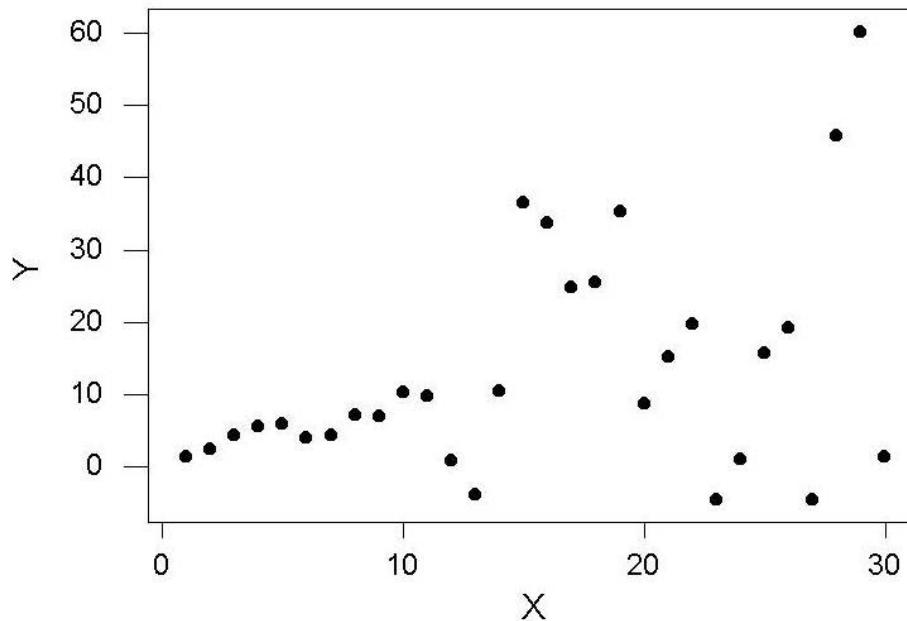
If there is a U-shaped pattern, the relationship is non-linear:



and so **we should not be fitting a straight line** to data.

Scary patterns in residual vs fits plots

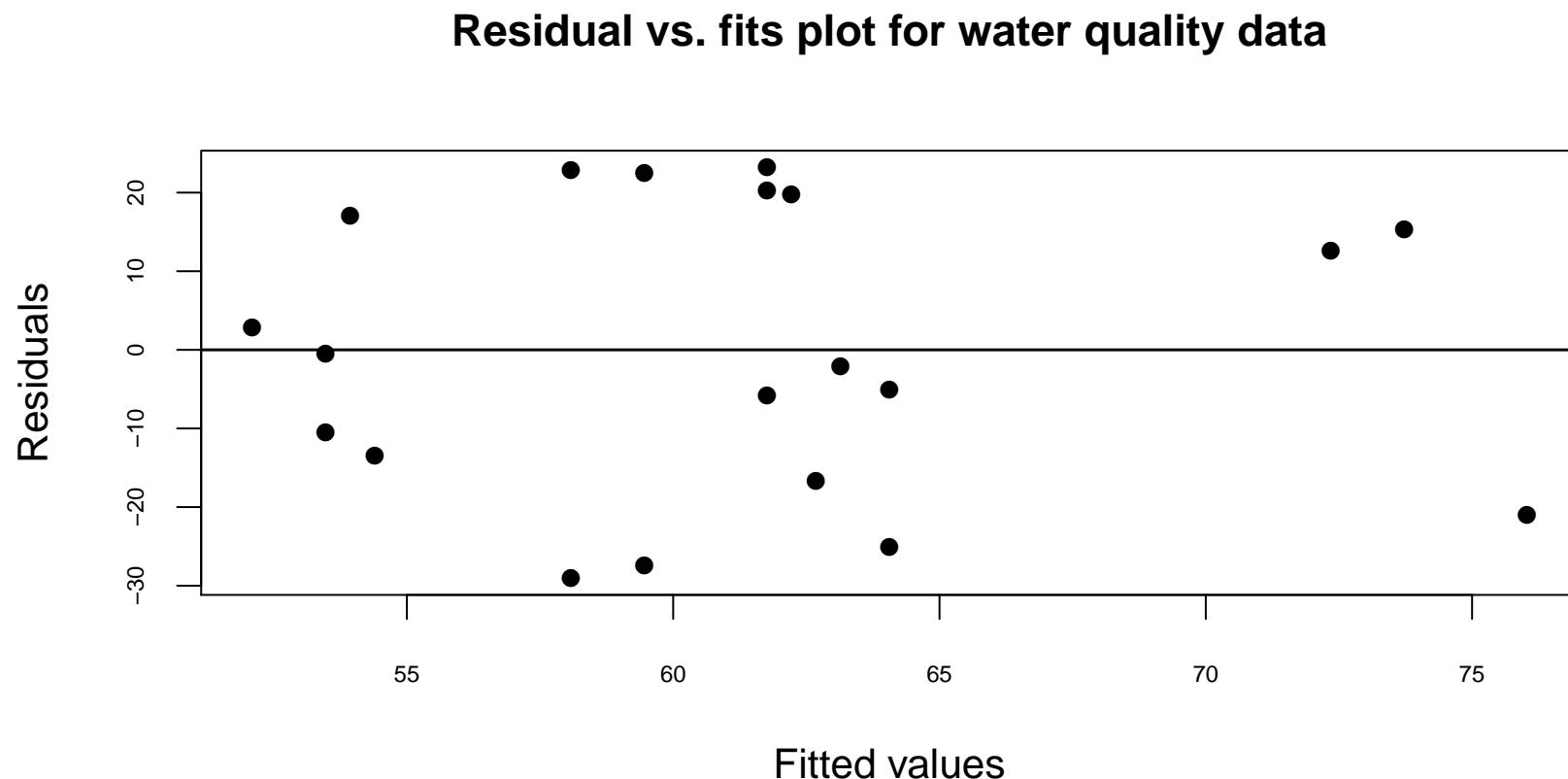
If there is a fan-shaped pattern, the variance changes with x :



and so we should not be assuming that the errors from the line have the same spread for all x .

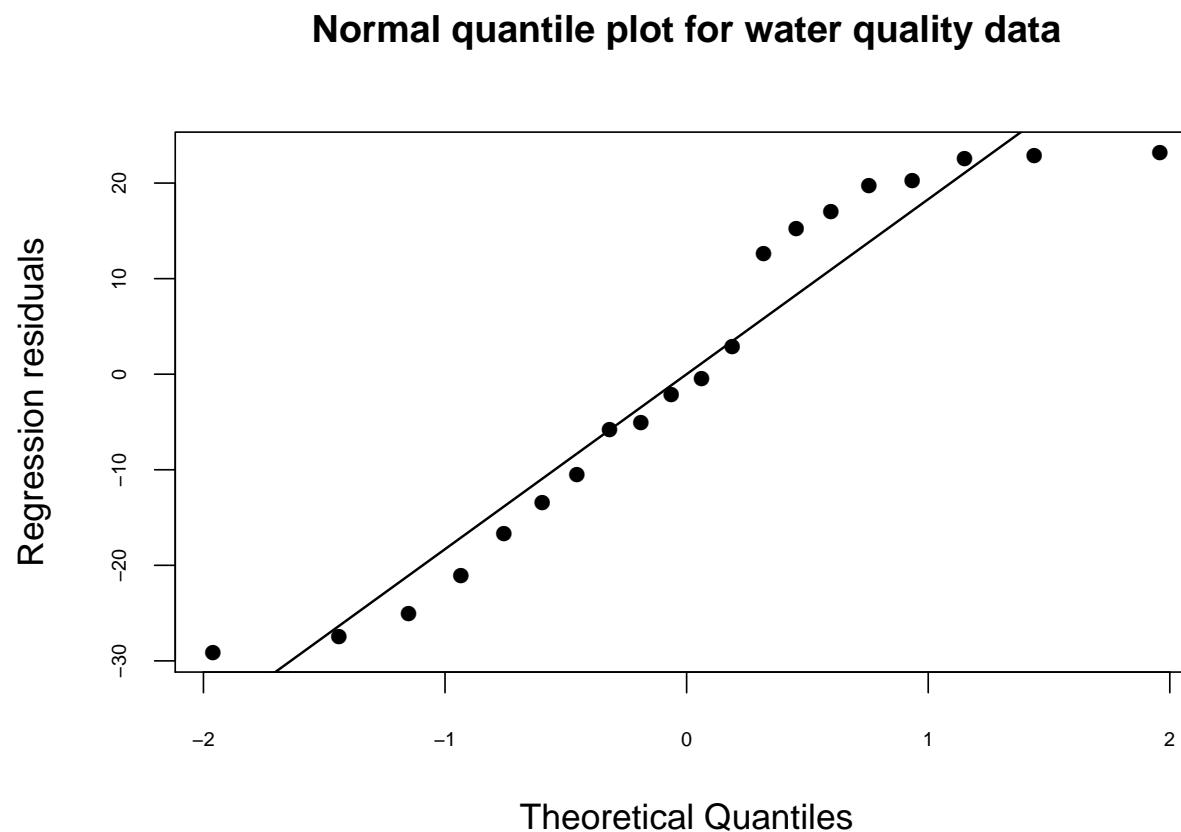
The water quality example: residual vs fits plot

What does this plot suggest concerning linear regression model assumptions? (and which ones?)



The water quality example: normal quantile plot

What does this plot suggest concerning linear regression model assumptions? (and which ones?)



Influential observations

An influential observation is one which has an **unusual x -value**. These are often easily seen in residual vs fits plots, although it gets more complicated for multiple linear regression (coming later today).

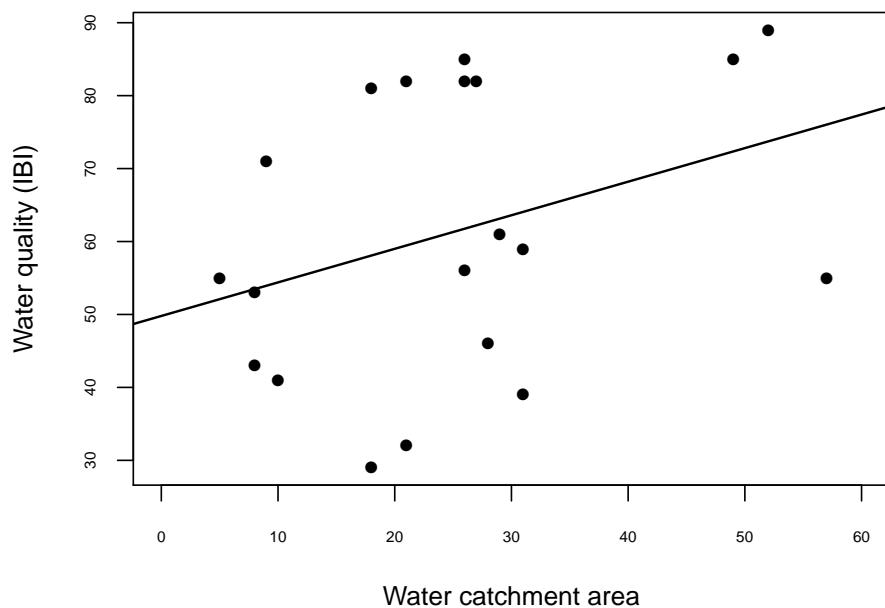
Influential observations are dangerous because they have **undue influence** on the fitted line – pretty much the whole fit can come down to the location of one point.

Once detected, a simple way to see if the whole story comes down to these influential values is to remove them and seeing if this changes anything. If it doesn't, nothing to worry about.

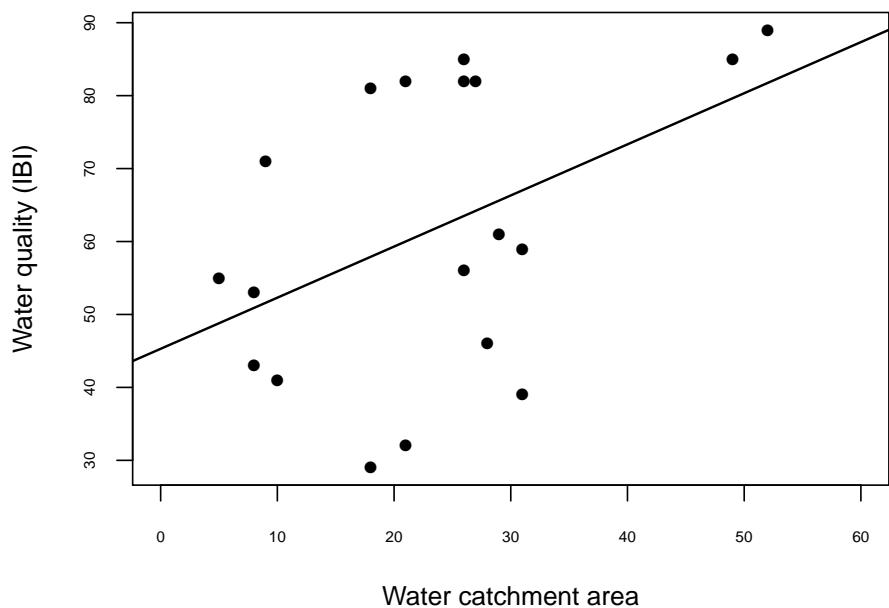
High influence points can often be avoided by transformation of the x variable, e.g. $\log(x)$, \sqrt{x} .

The water quality example: slightly influential observation

Scatterplot of water quality vs. catchment area



Wrecked scatterplot of water quality vs. catchment area



Notice the equation became a bit steeper on removal of the most extreme x -value (most influential point, not scary-influential though).

R^2 as proportion of variance explained

A nice way to summarise the strength of regression is the R^2 value, **the proportion of variance** in the y variable that has been **explained by regression** against x . For the water quality example:

Multiple R-squared: 0.118, Adjusted R-squared: 0.06898

so 11.8% of variance in water quality can be explained by catchment area.

But R^2 is a function of the sampling design as well as the strength of association, so is difficult to generalise

e.g. sampling a broader range of catchment sizes would increase R^2 , without changing the underlying water quality-catchment area relationship.

Equivalence of two-sample t -test and linear regression

The hardest part for today....

Equivalence of two-sample *t*-test and linear regression

Have a go at Session 2 exercise 1. (Guinea pigs and nicotine)

```
> datsmoke <- read.csv("smokePregnant.csv")
> t.test(datsmoke$Nicotine, datsmoke$Control, var.equal=TRUE)
```

Two Sample t-test

```
data: dat$Nicotine and dat$Control
t = 2.671, df = 18, p-value = 0.01558
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 4.460667 37.339333
sample estimates:
mean of x mean of y
        44.3       23.4
```

Note we can use linear regression to analyze the data... (!). First we reorganise the data a bit.

```
> errors=c(datasmoke$Nicotine,datasmoke$Control)
> treatment=rep(c("Nicotine","Control"),each=nrow(datasmoke))
> datasmoke2 <- data.frame(errors=errors,treatment=treatment)

> head(datasmoke2)
   errors treatment
1      38  Nicotine
2      26  Nicotine
3      33  Nicotine
4      89  Nicotine
5      66  Nicotine
6      23  Nicotine
```

Then we can use the `lm` function to fit a linear model.

```
> ft.smoke=lm(errors~treatment, data=datasmoke2)
> summary(ft.smoke)
```

Call:

```
lm(formula = errors ~ treatment, data = datAll)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	23.400	5.533	4.229	0.000504 ***
treatmentNicotine	20.900	7.825	2.671	0.015581 *

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 17.5 on 18 degrees of freedom

Multiple R-squared: 0.2838, Adjusted R-squared: 0.2441

F-statistic: 7.134 on 1 and 18 DF, p-value: 0.01558

Equivalence of two-sample t -test and linear regression

Two important things to notice:

- The regression intercept equals the sample mean of the Control group
- The P -value from the t -test and from the regression slope are the same.

But **why???**

The regression line goes through the sample means

Recall that the regression line is estimated by least squares.

For a single sample, **the sample mean is the least squares estimate** of the centre of the sample.

For two samples (lined up along the x -axis), a line will need to go through each sample mean if it wants to be the least squares estimate.

So simple linear regression just joins the two sample means.

What does this mean for the slope and elevation?

In our regression analysis, # of errors is the y -variable and treatment is the x -variable.

Now there are only two values on the x -axis (Control and Nicotine). R puts Control values at $x = 0$, and Nicotine values at $x = 1$.

This means:

- The y -intercept is the Control mean.
- The slope is the mean difference between Nicotine and Control.

Useful graph:

The t -test as a linear model

We can write the two-sample t -test as a linear model. Recall that a simple linear regression model has the form:

$$\mu_y = \beta_0 + \beta_1 x$$

Whereas the two-sample t test uses the model:

$$\mu_y = \begin{cases} \mu_{\text{Control}} & \text{for subjects in the Control group} \\ \mu_{\text{Nicotine}} & \text{for subjects in the Nicotine group} \end{cases}$$

If we let $x = 0$ for Control and $x = 1$ for Nicotine then the two-sample t -test is exactly the regression model with:

$$\begin{aligned}\beta_0 &= \mu_{\text{Control}} \\ \beta_1 &= \mu_{\text{Nicotine}} - \mu_{\text{Control}}\end{aligned}$$

So testing H_0 : “slope is zero” can test H_0 : “means are equal”!

Recall the two-sample t -test assumptions:

1. The y -values are **independent** in each sample.
2. The y -values are **normally distributed with constant variance**.

$$y_{ij} \sim \mathcal{N}(\mu_i, \sigma^2)$$

And the simple linear regression assumptions:

1. The observed y values are **independent**, after accounting for x .
2. The y values are **normally distributed with constant variance**.

$$y_i \sim \mathcal{N}(\mu_i, \sigma^2)$$

3. There is a **straight line relationship** between mean of y and x

$$\mu_i = \beta_0 + \beta_1 x_i$$

These are the **same set of assumptions**.

They can be **checked the same way**.

(The linearity assumption is not important for two-sample t , as we only have two points on the x -axis.)

What does this all mean?

So now you don't need to worry about two-sample t -tests any more – just think of it as linear regression with a binary predictor variable.

There is just one more thing we need to talk about – multiple regression (next session). After that you can handle pretty much any (fixed effects) sampling design **under a single framework**. Fitted using the same function, interpreted in the same way.

Distribution of the response is what matters

Recall that we originally had the following rules for when to use two-sample t vs regression:

Two-sample t : when y is quantitative and x is binary

Linear regression: when y is quantitative and x is quantitative

The fact that two-sample t = linear regression means that **it doesn't matter what type of variable x is.**

When choosing an analysis method, we don't need to know about x (the “predictor”), we just need to worry about y (the “response”).

If y is quantitative, try a linear regression (“linear model”).

We will worry about other responses types tomorrow.

Why don't we have to worry about the distribution of the predictor?

Because regression **conditions** on x . A regression model says:
“If x was this value, what would we expect y to be?”

The thing we are treating as random is y . The x value has been given in the question, we don't need to treat it as random (even if it is).

But don't ignore x completely...

We don't have to worry about the distribution of x when choosing an analysis method... but we should worry about it when actually doing the analysis.

If x is strongly skewed then you get high influence points (bad news) – so if x is quantitative, it is best to look at its distribution and consider transforming it for analysis if it is strongly skewed.

So don't ignore the distribution of x completely, **just don't use it to decide on an analysis method.**

Doing a `pairs` plot of all x variables before using them in analysis is a good idea – check if you should transform them to reduce skew/outliers.

Linear Models

- Multiple regression
- Analysis of variance (ANOVA)

(But it's all just the same linear model...)

Multiple regression

Example – Global plant height

Angela collected some data on how tall plants are at lots of different places around the world. Can latitudinal variation in plant height be explained by climate?

What does the question tell us – descriptive, interval estimation, hypothesis testing, ...

What do the data tell us – one variable or more? What type of response?

What sort of analysis method are you thinking?

Multiple regression model

Multiple linear regression is a special name for linear models that have **more than one x variable** that we want to use to predict y .

It's an extension of simple linear regression to two or more x variables.

The multiple regression model for two predictor variables x_1 and x_2 is:

$$\begin{aligned}y &\sim \mathcal{N}(\mu_y, \sigma^2) \\ \mu_y &= \beta_0 + \beta_1 x_1 + \beta_2 x_2\end{aligned}$$

The model for the mean can be written in vector notation as:

$$\mu_y = \beta_0 + \mathbf{x}^T \boldsymbol{\beta}$$

where $\boldsymbol{\beta}$ and \mathbf{x} are vectors.

Geometrically, this fits a plane in three dimensions, not a line (in 2D).

Assumptions of multiple regression

1. The observed y values are **independent**, after conditioning on x .
2. The y values are **normally distributed** with **constant variance**

$$y \sim \mathcal{N}(\mu_y, \sigma^2)$$

3. There is a **straight line relationship** between mean of y and each x variable

$$\mu_y = \beta_0 + \mathbf{x}^T \boldsymbol{\beta}$$

Look familiar? Same assumptions as simple linear regression, checked the same way as before (residual vs fits plots, normal quantile plots).

It can also be useful to plot residuals against each x variable to check that each is linearly related to μ_y .

Multiple regression in R – same same

You fit multiple regression models the same way as simple linear regression. The output looks the same too! Simple linear regression against lat:

```
> datheight <- read.csv("plantHeightSingleSpp.csv")
> ft.height1=lm(height~lat, data=datheight)
> summary(ft.height1)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)		
(Intercept)	14.42516	1.72433	8.366	1.78e-14 ***		
lat	-0.17631	0.04847	-3.637	0.000362 ***		

Signif. codes:	0 ***	0.001 **	0.01 *	0.05 .	0.1	1

Residual standard error: 10.95 on 176 degrees of freedom

Multiple R-squared: 0.06991, Adjusted R-squared: 0.06463

F-statistic: 13.23 on 1 and 176 DF, p-value: 0.0003617

Now a multiple linear regression against rain and lat:

```
> ft.height2=lm(height~rain+lat, data=datheight)
> summary(ft.height2)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.521724	3.077683	1.469	0.143575
rain	0.004058	0.001062	3.823	0.000183 ***
lat	-0.034109	0.059707	-0.571	0.568547

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 10.55 on 175 degrees of freedom
Multiple R-squared: 0.1416, Adjusted R-squared: 0.1318
F-statistic: 14.43 on 2 and 175 DF, p-value: 1.579e-06

New ideas in multiple regression

The maths, and the computation, are pretty much the same as for simple linear regression. But there are a few new ideas to look out for when you have multiple x values:

1. Interpret coefficients as conditional effects (not marginal)
2. Plot partial residuals to visualise conditional effects
3. Testing of multiple slope parameters
4. Multi-collinearity affects power

1. Interpret as conditional effects

Multiple regression coefficients should be interpreted **conditionally on all other predictors** in the model.

e.g. In the multiple regression, the coefficient of `lat` tells us the effect of `lat` after controlling for the effect of `rain`. That is, what is the association of latitude with height, **after controlling for the effect of rain?**

Recall that in simple linear regression, the slope was $\hat{\beta}_{\text{lat}} = -0.17$ (and significant). After controlling for the effect of `rain`, the slope was much flatter $\hat{\beta}_{\text{lat}} = -0.004$ (and not significant).

This means that, if assumptions are satisfied, most of the association between latitude and height is explained by precipitation.

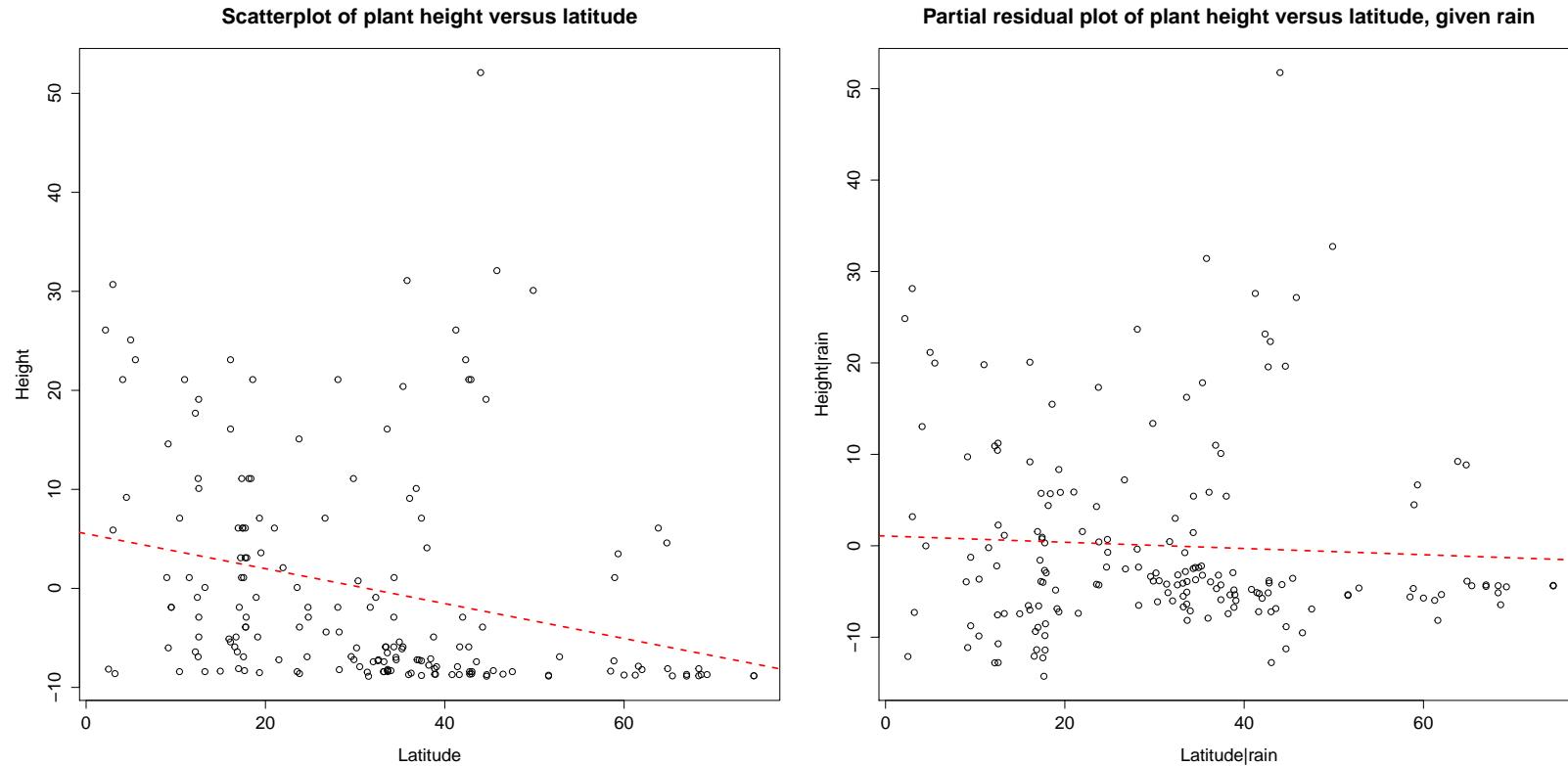
2. Partial residual plots

Partial residual plots let us look at the effect of a variable after controlling for another. (“component plus residual plot”, `crPlots` from `car` package.)

This is a good way of **visualising the conditional effect of one variable given another.**

(Also handy as an assumption check – do you think assumptions are reasonable?)

Partial residual plot for global plant height:



```
> library(car)
> crPlots(ft.height1, terms = ~lat, xlab="Latitude", ylab="Height",
grid=FALSE, smooth = FALSE) ##left plot
> crPlots(ft.height2, terms = ~lat, xlab="Latitude|rain",
ylab="Height|rain", grid=FALSE, smooth = FALSE) ## right plot
```

3. Testing of multiple slope parameters

Notice the bottom line in the output on slide 3.6:

F-statistic: 14.43 on 2 and 175 DF, p-value: 1.579e-06

This tests the null hypothesis that y is unrelated to **any** x variables.

What do you conclude?

What if we want to know

Does latitude explain the effect of climate on plant height?

We will use annual precipitation and mean temperature as “climate”.
To answer this question, we want to compare two models:

$$\mu_{\text{height}} = \beta_0 + \text{lat} \times \beta_{\text{lat}} + \text{temp} \times \beta_{\text{temp}} + \text{rain} \times \beta_{\text{rain}} \quad (1)$$

$$\mu_{\text{height}} = \beta_0 + \text{lat} \times \beta_{\text{lat}} \quad (2)$$

We want a **single test** of the hypothesis that there is no effect of `rain` nor `temp` (*i.e.* $H_0 : \beta_{\text{temp}} = \beta_{\text{rain}} = 0$), while accounting for the fact that there is a relationship with latitude.

Tests of multiple parameters on R— the anova function

To test model (1) against model (2), we can use the anova function:

```
> ft.Lat=lm(height~lat, data=datheight)
> ft.LatClim=lm(height~lat+rain+temp, data=datheight)
> anova(ft.Lat,ft.LatClim)
```

Analysis of Variance Table

Model 1: height ~ lat

Model 2: height ~ lat + rain + temp

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	176	21094				
2	174	19457	2	1637.2	7.3206	0.000886 ***

	Signif. codes:	0	***	0.001	**	0.01 * 0.05 . 0.1 1

How do you interpret these results?

Note on tests of multiple parameters

The above approach only works for **nested models** – when one model includes all the terms in the other model plus some extra ones (“contained in the larger model”)

For instance, you can't use the `anova` function to compare:

$$\mu_{\text{height}} = \beta_0 + \text{temp} \times \beta_{\text{temp}} + \text{rain} \times \beta_{\text{rain}} \quad (3)$$

$$\mu_{\text{height}} = \beta_0 + \text{lat} \times \beta_{\text{lat}} \quad (4)$$

All the terms in model (4) would need to be found in model (3) also, in order to use `anova` to compare the two models.

4. Multi-collinearity

Multi-collinearity is where some of the predictor variables are highly correlated.

It is a problem when making inferences about coefficients, as the **standard errors become inflated**. (So confidence intervals for slope parameters are wider, and P -values are larger.)

(Not such a problem for prediction, or for global tests.)

e.g. Adding rainfall in the wettest month as a predictor, as well as annual precipitation, we obtain

```
> ft.climproblems=lm(height~rain+rain.wetm+lat, data=datheight)
> summary(ft.climproblems)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3.326736	3.314338	1.004	0.317
rain	0.002006	0.002362	0.850	0.397
rain.wetm	0.016613	0.017082	0.973	0.332
lat	-0.013176	0.063477	-0.208	0.836
...				

Notice the P -value for rain (and all covariates) is no longer significant.

A common way to check for multi-collinearity is to compute **variance inflation factors** for each explanatory variable x . These tell us the factor by which standard errors are larger than they would have been if explanatory variables were uncorrelated. VIFs near 1 are good, values in the 5-10 range are ringing some alarm bells.

```
> library(car)
> vif(ft.height2)
      rain      lat
1.634535 1.634535
> vif(ft.climproblems)
      rain rain.wetm      lat
8.086291 9.122699 1.846891
```

As before, adding `rain.wetm` to the model already containing `rain` is problematic (but to `rain` only, `lat` is still OK).

Multi-collinearity

Another way to see what is going on is to simply **look at the correlation** between predictor variables,

```
> X = data.frame(datheight$lat,datheight$rain,datheight$rain.wetm)
> cor(X)

            datheight.lat   datheight.rain  datheight.rain.wetm
datheight.lat      1.0000000 -0.6230611 -0.6765425
datheight.rain     -0.6230611  1.0000000  0.9360246
datheight.rain.wetm -0.6765425   0.9360246  1.0000000
> pairs(X)
```

The pairs plot and the correlation matrix suggest that `rain` and `rain.wetm` are highly positively correlated.

Analysis of variance

Alistair looked at the density (per gram of seaweed) of invertebrate epifauna settling on seaweed patches with different levels of isolation (0, 2, or 10 metre buffer) from each other. He wants to know: Does invertebrate density change with isolation?

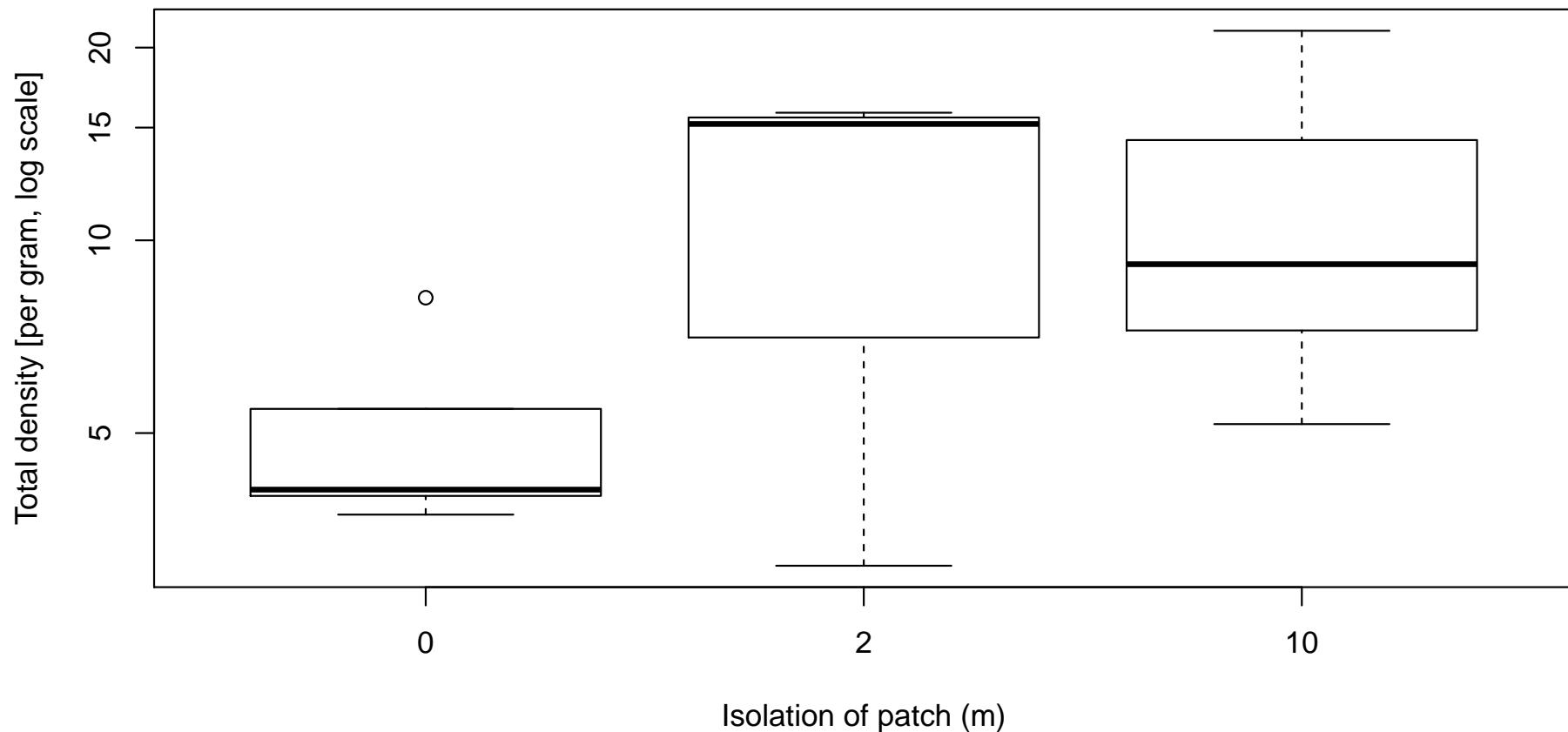
What does the research question ask us – descriptive, estimation, hypothesis testing, etc?

What do the data tell us – one variable or two? What type of response variable?

What graph would you use to visualise the data?

So how would you analyse the data?

Comparative histogram of total density



Response variable – density, quantitative.

(So we can use some type of linear model.)

Predictor variable – isolation (0, 2, or 10). This is **categorical** with three levels.

The analysis method for this situation is often referred to as **analysis of variance (ANOVA)**. But it is just another example of a linear model...

Equivalence of ANOVA and multiple regression

Recall that a two-sample t -test was just linear regression with a binary predictor variable.

Multiple regression is an extension of simple linear regression, and ANOVA is an extension of the two-sample t -test.

It turns out that ANOVA is a special case of multiple regression (linear models).

Recall that a multiple regression equation with two predictors has the form:

$$\mu_y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

Analysis of variance uses the model (for **the seaweed example**):

$$\mu_y = \begin{cases} \mu_0 & \text{for subjects in the Isolation= 0 group} \\ \mu_2 & \text{for subjects in the Isolation= 2 group} \\ \mu_{10} & \text{for subjects in the Isolation= 10 group} \end{cases}$$

If we let $(x_1 = x_2 = 0)$ for Isolation= 0, $(x_1 = 1, x_2 = 0)$ for Isolation= 2, let $(x_1 = 0, x_2 = 1)$ for Isolation= 10, then the analysis of variance model follows the regression model with:

$$\mu_0 = \beta_0$$

$$\mu_2 = \beta_0 + \beta_1$$

$$\mu_{10} = \beta_0 + \beta_2$$

Assumptions of ANOVA

Similar to multiple regression:

1. The observed y values are **independent**, conditional on x .
2. The y values are **normally distributed with constant variance**

$$y \sim \mathcal{N}(\mu_y, \sigma^2)$$

But we don't have to worry about the **linearity** assumption (because we only observe data at two points on each x -axis).

How do we check the above assumptions?

ANOVA output in R

You can fit ANOVA the same way as linear regression, just **make sure your x variable is a factor**.

```
> dathabconf <- read.csv("HabitatConfig.csv")
> dathabconf$Dist = factor(dathabconf$Dist)
> ft.habconf=lm(Total~Dist, data=dathabconf)
> anova(ft.habconf)
```

Analysis of Variance Table

Response: Total

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Dist	2	300.25	150.123	8.5596	0.0005902 ***
Residuals	54	947.08	17.539		

Any evidence that distance of isolation is related to density?

What are the assumptions made? How can they be checked?

Multiple comparisons

When doing an ANOVA with more than two levels of a factor, the next question is: which factor levels differ from which?

e.g. Recall Alistair looked at the density (per gram of seaweed) of invertebrate epifauna settling on seaweed patches with different levels of isolation (0, 2, or 10 metre buffer) from each other. He wants to know: Does invertebrate density change with isolation?

Here are his ANOVA results:

```
> dathabconf$Dist = factor(dathabconf$Dist)
> ft.habconf=lm(Total~Dist,data=dathabconf)
> anova(ft.habconf)
```

Analysis of Variance Table

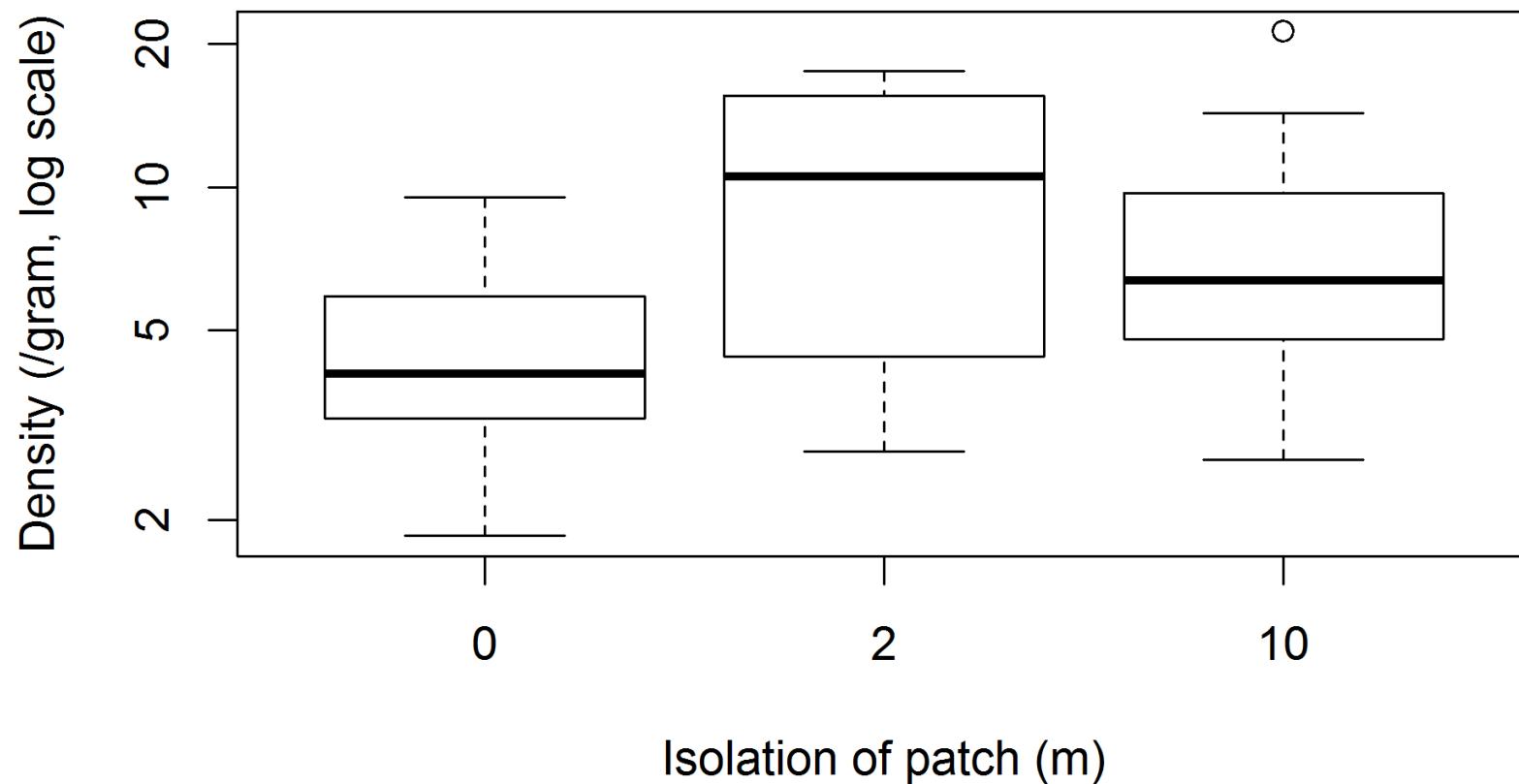
Response: Total

Df	Sum Sq	Mean Sq	F value	Pr(>F)
Dist	2	300.25	150.123	8.5596 0.0005902 ***
Residuals	54	947.08	17.539	

But where are the differences: between 0 and 2, or between 0 and 10, or between 2 and 10? Or some combination thereof?

We could start by plotting the data...

```
> plot(databconf$Total~databconf$Dist, log="y",
xlab="Isolation of patch (m)", ylab="Density (/gram, log scale)")
```



Need more than just confint

If we start by running `confint` on the fitted model we don't get what we want:

```
> confint(ft.habconf)
2.5 % 97.5 %
(Intercept) 2.7785049 6.533423
Dist2       2.9211057 8.460686
Dist10      0.4107071 5.720963
```

This compares each of the “2” and “10” groups to “0”.

But:

- What about “2” vs “10”?
- When doing multiple tests we should correct for this in assessing significance.

Multiple comparisons in R

You can do multiple comparisons on R using the TukeyHSD or pairwise.t.test functions, but you can get nicer results and more flexibility using the multcomp library.

First time you use this R library on your computer, you will need to install it :

```
install.packages("multcomp")
```

R libraries offer additional functions for special purposes that are not available in the base download. There are thousands of libraries. It is easy to contribute your own and it is publicly available in days – but noone checks them centrally for correctness, corrections will only come from users. Hence R is kind of like the Wikipedia of statistics.

```
> library(multcomp)
> contDist = mcp(Dist="Tukey") # telling R to compare on the Dist factor
> compDist = glht(ft.habconf, linfct=contDist) # now run multiple comparisions
> summary(compDist) # present a summary of the results
```

Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts

Fit: lm(formula = Total ~ Dist, data = habconfig)

Linear Hypotheses:

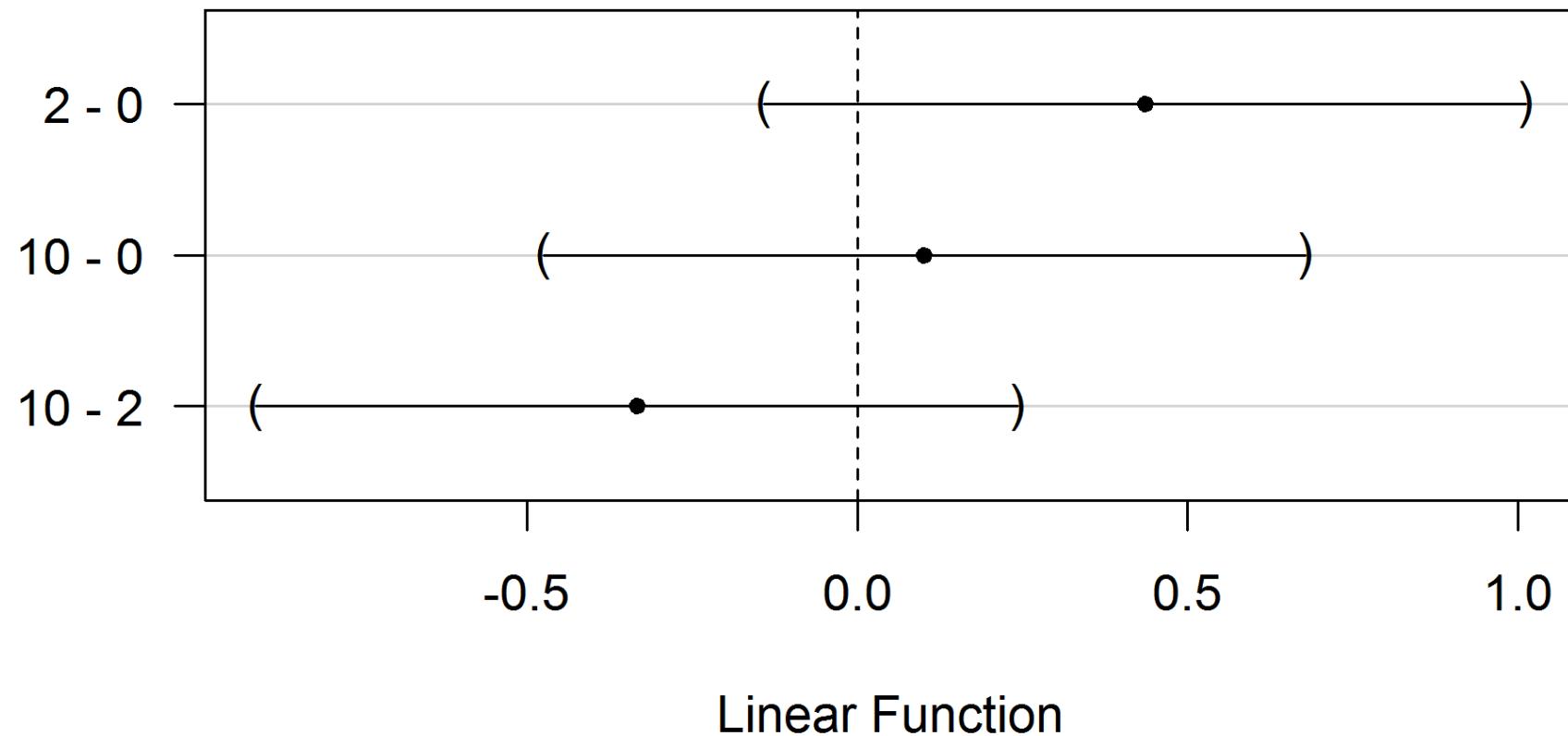
	Estimate	Std. Error	t value	Pr(> t)
2 - 0 == 0	5.691	1.382	4.119	<0.001 ***
10 - 0 == 0	3.066	1.324	2.315	0.0623 .
10 - 2 == 0	-2.625	1.382	-1.900	0.1483

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1
(Adjusted p values reported -- single-step method)

You can also use the confint function on your results, which have been stored as compDist. Or plot...

```
plot(compDist)
```

95% family-wise confidence level



So what can we conclude is different from what?

What is this simultaneous test thing about?

Every time you do a hypothesis test, and conclude you have significant evidence against H_0 when $P < 0.05$, you have a 5% chance of accidentally rejecting the null hypothesis (“Type I error”).

If doing three pairwise comparisons (2 – 0, 10 – 0, 10 – 2) then this would give about a 15% chance of accidentally declaring significance. (The more groups to compare, the more the chance – with 10 different treatment groups you are almost guaranteed a false positive!)

The `multcomp` package (and TukeyHSD, Tukey’s “Honestly Significant Difference”, and `pairwise.t.test`) are more conservative to account for this, so that overall across all comparisons, we maintain a 5% chance of accidentally declaring significance.

We are correcting for “multiple testing”. `mcp` stands for `multiple comparisons procedure`.

Weirder linear models

- Paired and blocked designs
- ANCOVA
- Factorial experiments
- Interactions in regression

It's all just linear models!

Paired and blocked designs

Richard went bird-counting in 12 transects – in 1992, and 2012. He wants to know: is there evidence of a change in bird counts across the two sampling times?

Site:	CST	DAR	GBT	GCG	GRC	LBC	LPC	MST	PCO	PEL	SGU	TRN
1992:	47	13	13	0	46	222	104	35	110	969	74	0
2012:	0	61	0	0	58	95	0	0	432	1068	26	0

What does the research question tell us – descriptive, estimation, hypothesis testing...

What do the data tell us – one variable or more? What type of response?

So how would you analyse the data?

Any problems with assumptions?

Paired data violate the independence assumption

Independence of observations is violated by sampling designs with **paired data** – if the count was high in 1992, then it is likely to be high in 2012 also.

Common approach to analysis of paired data: **calculate paired differences and analyse those**.

1992 counts - 2012 counts:

47 -48 13 0 -12 127 104 35 -322 -99 48 0

If $\mu_{1992} = \mu_{2012}$ then $\mu_{\text{difference}} = 0$. So we test if there is evidence that this sample has a non-zero mean.

We no longer assume independence across sites and sampling times – instead we assume **differences** are independent across sites (and normally distributed with constant variance).

In R, we could use the t.test function:

```
> datbird <- read.csv("birdCountYear.csv")
> t.test(datbird$X1992,datbird$X2012, paired=TRUE)
```

Paired t-test

```
data: dat$X1992 and dat$X2012
t = -0.2664, df = 11, p-value = 0.7948
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-82.57346 64.74012
sample estimates:
mean of the differences
-8.916667
```

Is there evidence of a change in bird counts across the two sampling times?

Paired *t*-test as a main effects ANOVA

Another way to think of the data is like this:

year	site	count
1992	CST	47
1992	DAR	13
1992	GBT	13
1992	GCG	0
1992	GRC	46
:	:	:
2012	CST	0
2012	DAR	0
2012	GBT	0
2012	GCG	0
2012	GRC	58
:	:	:

And you could just **fit a linear model** with terms for year and site.

Here is some code to do this in R using the `melt` function in the `reshape` package.

```
> colnames(datbird) <- c("site", "1992", "2002", "2012")
> datbirdLong <- melt(datbird)
Using site as id variables
> colnames(datbirdLong)[c(2,3)] <- c('year','count')
> head(datbirdLong)
site year count
1 CST 1992    47
2 DAR 1992    13
3 GBT 1992    13
4 GCG 1992     0
5 GRC 1992    46
6 LBC 1992   222
```

```
> ft.bird=lm(count~site+year,data=subset(datbirdLong,year!="2002"))
> anova(ft.bird)
Analysis of Variance Table
```

Response: count

Df	Sum Sq	Mean Sq	F value	Pr(>F)	
site	11	1818119	165284	24.597	3.724e-06 ***
year	1	477	477	0.071	0.7948
Residuals	11	73915	6720		

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1					

The test of significance of the year term is **the same** as for the paired *t*-test. These methods are **equivalent**. Note that the *P*-values are the same!

What does this all mean? \Rightarrow Blocked designs

The above result is handy because by thinking of the paired setting as a linear model, we can see how to handle more complicated situations like this one:

Richard went bird-counting in 12 transects – in 1992, 2002, and 2012.

He wants to know: Is there evidence of a change in bird counts across the three sampling times?

site	CST	DAR	GBT	GCG	GRC	LBC	LPC	MST	PCO	PEL	SGU	TRN
1992	47	13	13	0	46	222	104	35	110	969	74	0
2002	60	36	0	0	21	35	0	12	3108	932	177	2
2012	0	61	0	0	58	95	0	0	432	1068	26	0

Assumptions of blocked design

1. The observed y values are **independent** given x – in other words, after accounting for site and year (see bottom)
2. The y values are **normally distributed with constant variance**

$$y_{ij} \sim \mathcal{N}(\mu_{ij}, \sigma^2)$$

3. The treatment effect is **additive** across blocks. For bird counts:

$$\mu_{ij} = \beta_{\text{site}i} + \beta_{\text{year}j}$$

(“Additive effects” = Linearity)

Check assumptions the same way.

Independence of y is **conditional on x** :

We **don't** assume abundances at each site and time are independent.
We do assume that **beyond** site and year, there are no further sources of correlation between observations.

You can think of this as a **blocked design**, where at each site (“block”) we get three measurements – one at each sampling time.

To analyse, we rearrange into three variables (count, year and site as before):

```
> ft.morebird=lm(count~site+factor(year), data=datbirdLong)
> anova(ft.morebird)
```

Analysis of Variance Table

Response: count

Df	Sum Sq	Mean Sq	F value	Pr(>F)
site	11	5792502	526591	2.2873 0.04746 *
factor(year)	2	404428	202214	0.8783 0.42955
Residuals	22	5064970	230226	

Signif. codes:	0 ***	0.001 **	0.01 * 0.05 . 0.1	1

Blocked Designs

The above can be considered as an example of a (randomised) blocks design.

For a more conventional example see:

<http://www.r-tutor.com/elementary-statistics/analysis-variance/randomized-block-design>

The point of blocking is to **control for known major sources of variability** that are not of direct interest to the research question (such as site-to-site variation).

The blocking factor is not of interest – so ignore its *P*-value in output!

Analysis of covariance (ANCOVA)

A *different* example: Alistair measured wet mass of algal beds as well, because that is expected to be important to epifauna density.

Is there an effect of distance of isolation after controlling for wet mass?

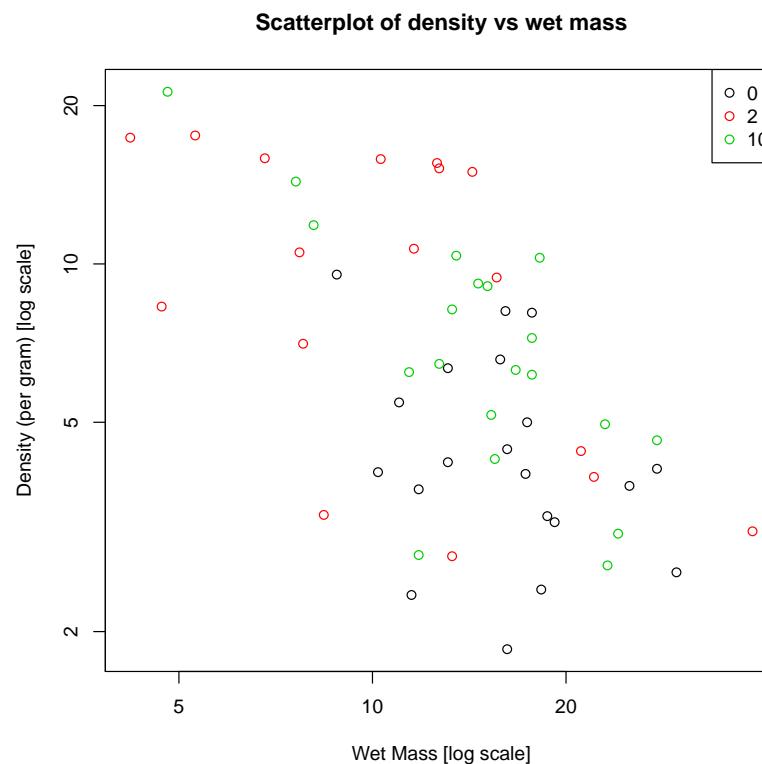
What does the research question tell us – descriptive, estimation, hypothesis testing...

What do the data tell us – how many variables? Distribution of the response variable?

What type of graph would you start with?

Useful plot on R:

```
> dathabconf <- read.csv("HabitatConfig.csv")
> plot(dathabconf$Total~dathabconf$Wmass, col=as.numeric(dathabconf$Dist),
log="xy", main="Scatterplot of density vs wet mass",
xlab="Wet Mass [log scale]", ylab="Density (per gram) [log scale]")
> legend("topright", levels(factor(dathabconf$Dist)), col=1:3, pch=1)
```



ANCOVA – it's just a linear model

This problem is similar to the randomised blocked design – we have a variable (`Wmass`) that we know is important but not of primary interest, i.e. we only want to control for this “covariate”. (Like `site` in the bird example.)

The difference is that the covariate is **quantitative** rather than being categorical. But that is no big deal for us – it is still a linear model! The code for analysis doesn't change, just the graphs.

```

> dathabconf$Dist=factor(dathabconf$Dist)
> ft.habconf=aov(Total~Wmass+Dist, data=dathabconf)
> summary(ft.habconf)

Df Sum Sq Mean Sq F value    Pr(>F)
Wmass           1  411.1   411.1  32.695 5.07e-07 ***
factor(Dist)   2  169.8    84.9   6.752  0.00244 **
Residuals     53  666.4    12.6
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1    1

```

(or you could use the `lm` function instead, but remember `aov` gives some nice multiple comparisons functions)

Assumptions of ANCOVA

Same same...

1. The observed y values are **independent**, conditional on x .
2. The y values are **normally distributed** with **constant variance**

$$y_{ij} \sim \mathcal{N}(\mu_{ij}, \sigma^2)$$

3. **linearity** – the effect of covariate on mean of y is **linear**, and effect of factor is **additive**. e.g. for epifauna data:

$$\mu_{ij} = \beta_{\text{Dist}i} + x_j \beta_{\text{Wmass}}$$

How can we check these assumptions?

Warning – order is important for both aov and anova

Notice results are different if you use:

```
> ft.difforder=aov(Total~Dist+Wmass,data=dathabconf)
> summary(ft.difforder)

             Df  Sum Sq Mean Sq F value    Pr(>F)
Dist          2   300.2  150.12   11.94 5.24e-05 ***
Wmass         1   280.7  280.67   22.32 1.74e-05 ***
Residuals    53   666.4   12.57
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1     1
```

Warning – order is important for aov and anova

The anova and aov functions use “Type I sums of squares” – they **sequentially add terms** to the model and test if each added term explains additional variation compared to those already in the model.

`lm(Total~Dist+Wmass,data=databconf)` then `anova(ft)` will fit the following sequence of models:

1. “Intercept model”, no terms for Dist or Wmass.
2. Dist only.
3. Dist and Wmass.

And in the anova call:

- Dist tests the first step (model 1 vs 2, any effect of Dist)
- Wmass tests the second step (model 2 vs 3, any additional effect of Wmass after controlling for Dist)

Alistair measured wet mass of algal beds as well, because that is expected to be important to epifauna density. He wants to know:

Is there an effect of distance of isolation after controlling for wet mass?

Which way should we specify the linear model?

...Wmass+Dist or ...Dist+Wmass?

drop1 for “Type II sums of squares”

Order matters for any linear model fitted on R – the exception being ANOVA with a balanced design.

But you can beat this “problem” using the `drop1` function:

```
> drop1(ft,test="F")
Single term deletions
```

Model:

Total ~ Dist + Wmass

Df	Sum of Sq	RSS	AIC	F value	Pr(>F)	
<none>		666.42	148.16			
Dist	2	169.81	836.22	157.09	6.7523	0.002442 **
Wmass	1	280.67	947.08	166.19	22.3213	1.737e-05 ***

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1						

Factorial experiments

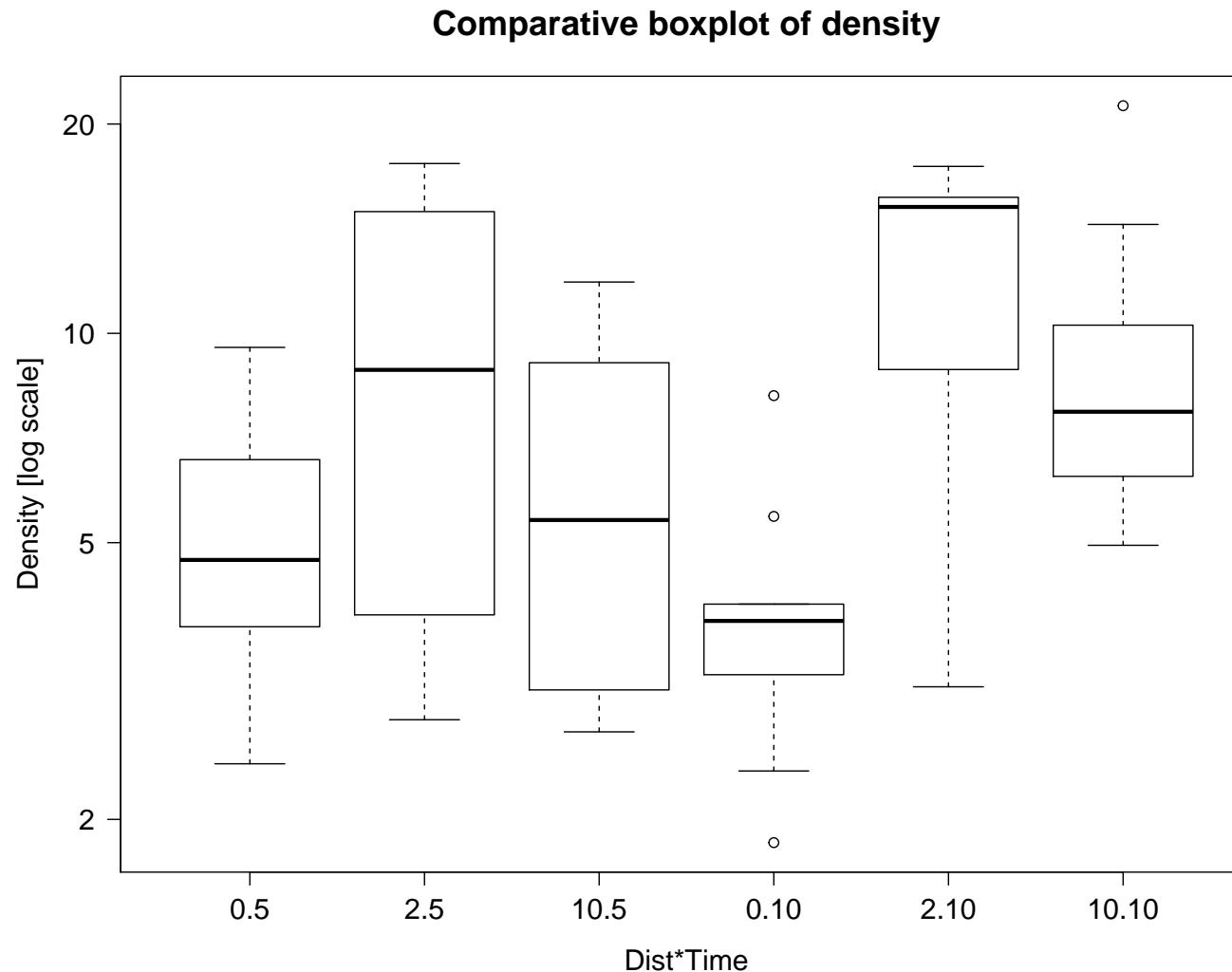
Alistair also looked at the density (per gram of habitat) of invertebrate epifauna settling on habitat patches over time periods of two different lengths (5 weeks and 10 weeks), as well as different isolation (0, 2, or 10 metre buffer). He wants to know: Does invertebrate density change with isolation? Does the isolation effect vary with time period?

What does the research question tell us – descriptive, estimation, hypothesis testing...

What do the data tell us – one variable or two, distribution of the response variable?

What type of graph would you start with?

```
> plot(dathabconf$Total ~ interaction(dathabconf$Dist,dathabconf$Time),  
log="y") ## and as usual use main, xlabel, ylabel to name axes
```



There are now two explanatory variables:

- Dist (of isolation): 0, 2 or 10 metres
- Time: 5 or 10 weeks

We have five replicate measurements at each combination of Dist and Time.

This is known as a **factorial design**.

It's just a linear model!

So how about this linear model?

```
> ft.habconf=lm(log(Total)~Dist+Time, data=databconf) ## Notice the log
```

But this assumes the isolation (`Dist`) effect is the same at each sampling time.

This doesn't answer the question:

Does the isolation effect vary with time period?

Factorial ANOVA **is** a linear model, but it's not this one.
We need an **interaction** term.

Interaction

An interaction between two variables tells us whether the nature of the effect of one variable **changes as the other variable changes**.

To answer the question “Does the isolation effect vary with time period?”, we need to test for an interaction between Dist and Time.

Often written as `Dist*Time`, but on R, as `Dist:Time`.

On R, `Dist*Time` is useful though as shorthand for “a factorial design with main effects and interactions”, i.e. `Dist + Time + Dist:Time`.

```

> ft.habconf2=aov(log(Total)~Time*Dist, data=dathabconf)
> summary(ft.habconf2)

      Df  Sum Sq Mean Sq F value    Pr(>F)
Time       1  0.243  0.2433   0.851  0.36055
Dist       2  5.032  2.5161   8.802  0.00052 ***
Time:Dist  2  1.467  0.7337   2.567  0.08668 .
Residuals 51 14.578  0.2859

---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

```

How would you interpret these results?

Hey why use Time*Dist, not Dist*Time?

What Df?

Degrees of freedom (“df”) for each term in the model tells you the number of (indicator) variables that were added to the model to account for that term.

For factors,

$$df = \# \text{ levels} - 1$$

e.g. `Time` has 1 df (two sampling times, 5 and 10).

`Dist` has 2 df (three distances, 0, 2, and 10)

For interactions, the rule is to multiply the degrees of freedom for the respective main effects e.g., `Dist:Time` has $2 \times 1 = 2$ df.

Why are the “Df” important?

They aren't! They used to be, back in the day, when they were used to manually compute stuff (variance estimates).

Df's are mostly useful just as a **check** to make sure nothing is wrong. In particular, if you accidentally treat a factor as quantitative, it will have 1 df no matter how many levels the factor has. If we forgot to turn Dist into a factor the table would look like this:

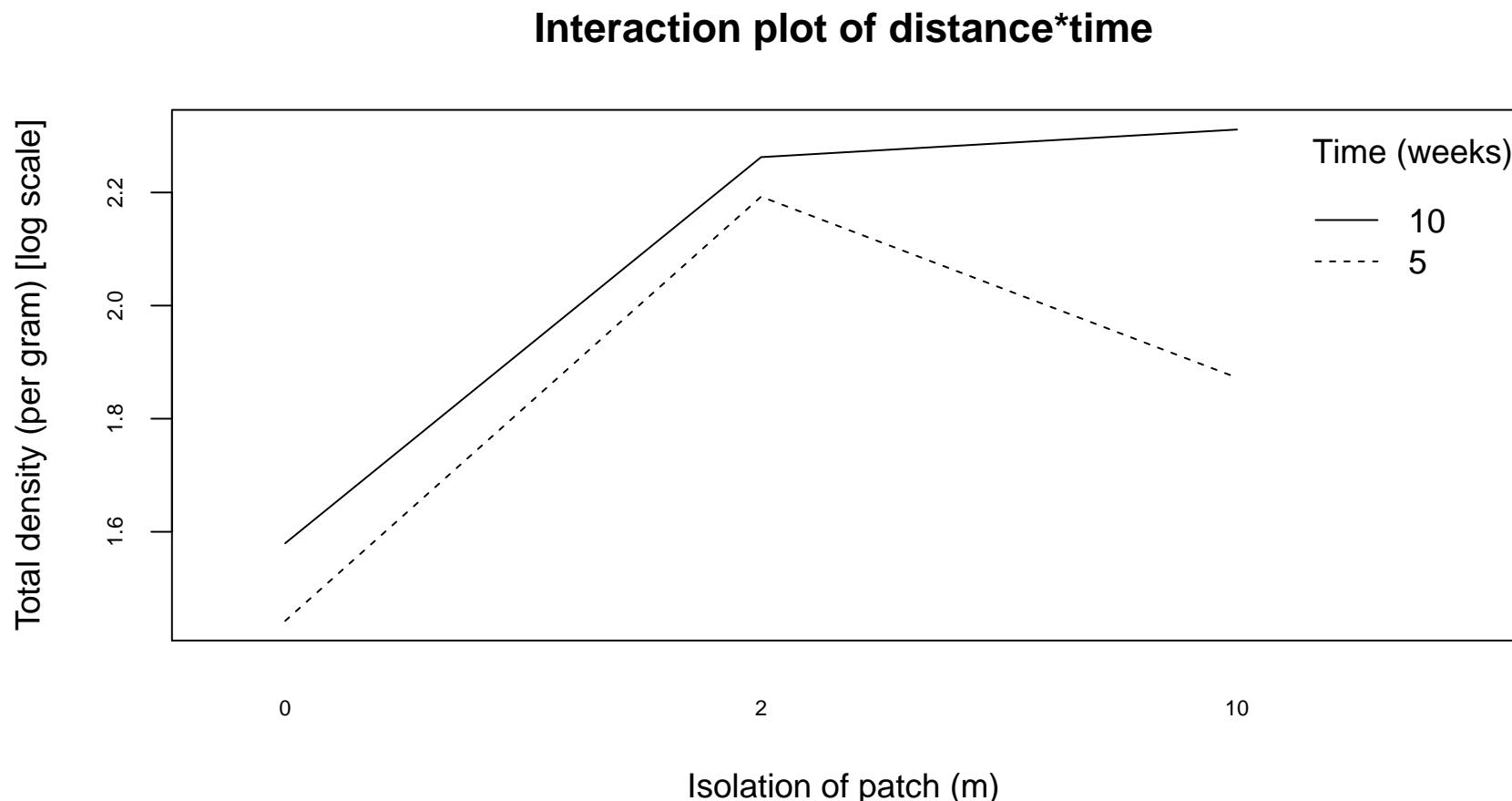
```
> ft.nofactor=aov(log(Total)~Time*Dist, data=dathabconf)
> summary(ft.nofactor)

             Df  Sum Sq Mean Sq F value Pr(>F)
Time          1  0.243  0.2433   0.667 0.4177
Dist          1  0.716  0.7164   1.964 0.1669
Time:Dist     1  1.030  1.0303   2.825 0.0987 .
Residuals    53 19.331  0.3647

---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1  1
```

Interaction plots

A helpful way to visualise interactions between factors is using an **interaction plot**.



```
interaction.plot(dathabconf$Dist, dathabconf$Time, fit.habconf2$fitted,  
xlab="Isolation of patch", ylab="Total density [log]", trace.label="Time")
```

Assumptions of Factorial Design

What assumptions are made in a factorial design linear model?

What can be done to check these assumptions are reasonable?

Interactions are cool

A lot of interesting ideas can be expressed as interactions.

Environmental impact (“BACI design”) – if we have monitoring data Before and After an impact, with Control and Impacted “treatments”, the treatment \times time interaction tells us if there was an environmental impact.

“Fourth corner” models – across several co-occurring species, if we model species abundance as a function of environment, spp traits, and the environment \times trait interaction, the interaction tells us how traits mediate different environmental response of different species.

Interactions are complicated

If Alistair has an interaction between Dist and Time, this means that there is no universal explanation for what is going on as Dist varies. So interactions complicate things.

The next step (if significant interaction) is to “break it down” – look at the effects of Dist separately for each sampling Time.

Multiple comparisons in factorial designs

You can use the same functions as before (TukeyHSD or pairwise.t.test) for main effects or interactions.

Tukey's comparisons for Dist:

```
> TukeyHSD(ft.habconf2, which="Dist")
  Tukey multiple comparisons of means
  95% family-wise confidence level
```

Fit: aov(formula = log(Total) ~ Time * Dist, data = dathabconf)

\$Dist

	diff	lwr	upr	p adj
2-0	0.7232463	0.29748692	1.1490056	0.0004293
10-0	0.4583808	0.05024677	0.8665148	0.0243203
10-2	-0.2648655	-0.69062483	0.1608939	0.2986241

Interactions in regression

Analysis of covariance

Recall that Alistair also measured wet mass of algal beds (AKA seaweed) as well, because that is expected to be important to epifauna density.

We assumed additivity – that distance of isolation had an additive effect on $\log(\text{density})$. Could there be an interaction between isolation and wet mass?

Graphically, an ANCOVA interaction would mean that the slope of the relationship between density (potentially log transformed) and Wmass changes as Dist changes.

We can test for interactions in analysis of covariance in just the same way as we test for interactions in factorial ANOVA.

```
> ft=aov(log(Total)~Wmass*Dist, data=dathabconf)
> summary(ft)

Df Sum Sq Mean Sq F value    Pr(>F)
Wmass          1  6.779   6.779   30.73 1.05e-06 ***
Dist           2  2.846   1.423    6.45  0.00318 **
Wmass:Dist    2  0.446   0.223    1.01  0.37122
Residuals     51 11.251   0.221
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Is there evidence of an interaction between the effects on density of wet mass and distance of isolation?

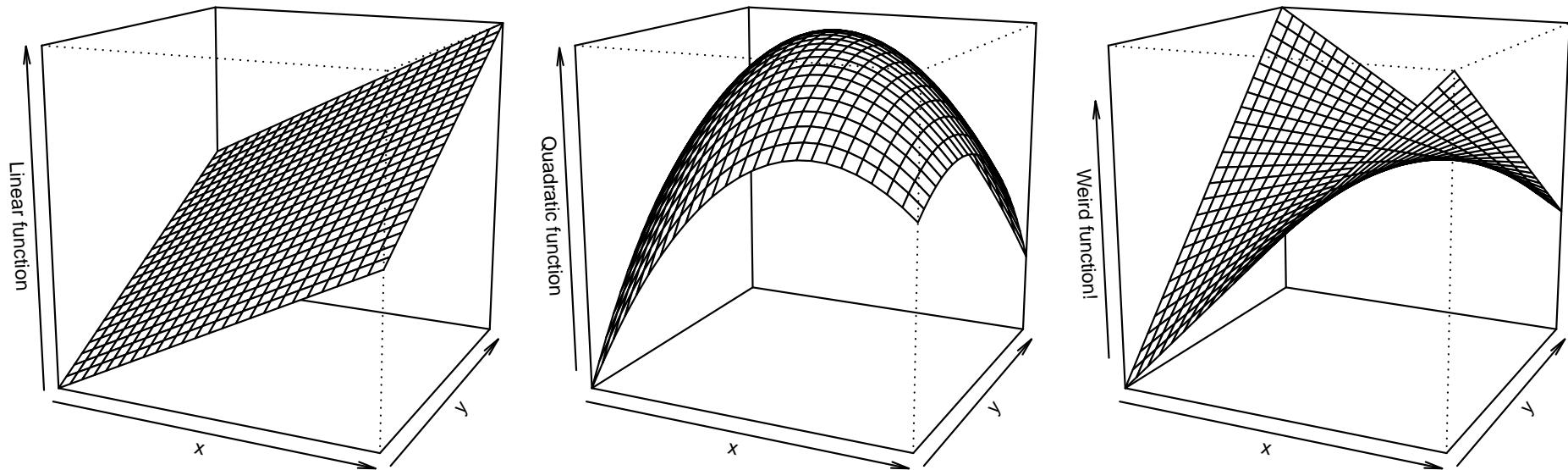
Interactions in multiple regression

In multiple regression models, you can also have **interactions between two continuous covariates** too, but it's a little tricky.

The interaction between two quantitative variables (`temp:rain`, say) is a **quadratic** term. Other quadratic terms are `temp^2` and `rain^2`.

It doesn't make sense to include interactions without including other quadratic terms also. It can lead to a really weird looking response surface...

Interactions without other quadratic terms look weird:



It doesn't make sense to include interactions between two quantitative variables ($x_1 : x_2$) without including other quadratic terms also (x_1^2, x_2^2).

Simplest way to fit quadratics on R is to use the poly function:

```
> datheight <- read.csv("plantHeightSingleSpp.csv")
> ft.height=lm(height~poly(cbind(rain,temp),degree=2), data=datheight)
> summary(ft.height)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	9.3929	1.2193	7.703	1e-12 ***	
poly(cbind(rain, temp), degree = 2)1.0	61.2940	15.6995	3.904	0.000136 ***	
poly(cbind(rain, temp), degree = 2)2.0	-19.9302	12.8228	-1.554	0.121955	
poly(cbind(rain, temp), degree = 2)0.1	-2.6263	14.7615	-0.178	0.858999	
poly(cbind(rain, temp), degree = 2)1.1	-156.1643	301.1911	-0.518	0.604784	
poly(cbind(rain, temp), degree = 2)0.2	0.9428	15.5539	0.061	0.951739	

Is there a significant interaction?

Model selection

- Why not R^2 ?
- Training/test data
- Cross-validation
- Information criteria
- Penalised estimation

Example - plant height and climate

Which climate variables best explain plant height?

Angela collects data on how tall plants are at lots of different places around the globe. She also has data on 19 different climate variables (precipitation and temperature). She is interested in how plant height relates to climate, and to which climate variables height relates most closely.

What does the question tell us – descriptive, hypothesis test, interval estimation, ...

What do the data tell us – one variable/more, what type of variable is the response

So how would you analyse the data?

The goal – model selection

The key difference here from what we have done previously is that we are primarily interested in choosing the best x variables (“to which climate variables height relates most closely”).

This is a **variable selection** or **model selection** problem – the goal is to select the best (or a set of best) models for predicting plant height.

Model selection as inference

So far we have talked about two types of statistical inference:

- Hypothesis testing – to see if data are consistent with a particular hypothesis
- Confidence interval estimation – constructing a plausible range of values for some parameter or key interest.

You can think of model selection as another type of inference.

Warning: Model selection gets very difficult very quickly

Consider a situation where you have a set of x variables and you want to fit all possible models (“all subsets”). If there are p variables, there are 2^p possible models – this gets unmanageable very quickly.

# variables	# models to fit
2	4
3	8
5	32
10	1,024
20	1,048,576
100	10^{30}
300	more than the number of electrons in the known universe!

If you have 200 observations and 10 variables, all-subsets is trying to choose from 1000+ models using just 200 observations, good luck!

This means two things:

- What the data says is the “best model” should be taken with a grain of salt. When there are lots of possible models, it is very hard for the data to make the right call.
- Simplify! The less candidate models you are comparing the better - don’t bother with anything you think is unrealistic, try to refine your question, do you really need all those variables?

Why not R^2 ?

What's the big deal – can't I just compare R^2 for all my candidate models, and pick the one with the best R^2 ?

The problem is accounting for **model complexity**. Making a model more complex (by adding more terms) can be a good thing but only if:

- The additional complexity is actually warranted
- You have enough data to do a good job of estimating the additional terms.

R^2 makes **no attempt** to account for the costs of model complexity – it keeps going up as you add more terms, even useless ones!

OK, why not use hypothesis tests?

Well why not add terms to the model which are significant, and remove terms when they are not significant?

This is commonly done but it is not a great idea:

- It is not what hypothesis testing was designed for.
- It has some undesirable properties (e.g. not “selection consistent”
 - is not guaranteed to pick the right model even when given enough data to do so)

Validation

The simplest way to compare predictive models is to see how well they predict to new data, **validation**. In the absence of new data, you can take a **test** or **hold-out** sample from the original data that is kept aside for model evaluation. The remaining **training** data are used to fit each candidate model.

It is critical that the test sample be **independent** of the training sample, otherwise this won't work. If all observations are independent (given x) then a random allocation of observations to test/training will be fine.

How to choose size of test sample?

There is no single best answer. However, one well-known argument (Shao 1993, 1997) is that as sample size n increases, the size of the training sample should increase, but as a proportion of n it should decrease towards zero. (This ensures selection consistency *i.e.* guaranteeing the correct model is chosen as n tends to infinity.)

An example strategy Shao (1993) suggested (which hence became a bit of a thing) is to use $n^{3/4}$ observations in the **training** sample. This can be quite harsh though:

n	$n^{3/4}$
20	9
50	19
100	32
200	53
1000	178

How to measure predictive performance?

For linear regression, the obvious answer is mean squared error:

$$\sum_{i=1}^{n_{\text{test}}} (y_i - \hat{\mu}_i)^2$$

where the summation is over test observations, for each of which we compare the observed y value, y_i , to the value predicted by the model fitted to the training sample, $\hat{\mu}_i$.

Comparing mean squared error for test data, for models with `temp` and one of `rain` and `rain.wetm`:

```
> datheight <- read.csv("plantHeightSingleSpp.csv")
> n = dim(datheight)[1]
> indTrain = sample(n,n^0.75) #select a training sample of size n^0.75:
> datTrain = datheight[indTrain,]
> datTest = datheight[-indTrain,]
> ft1 = lm(log(height)~temp+rain, dat=datTrain)
> ft2 = lm(log(height)~temp+rain.wetm, dat=datTrain)
> pr1 = predict(ft1, newdata=datTest)
> pr2 = predict(ft2, newdata=datTest)
> rss1 = mean( (log(datTest$height)-pr1)^2 )
> rss2 = mean( (log(datTest$height)-pr2)^2 )
> print( c(rss1,rss2) )
[1] 2.336706 2.399931
```

So it seems from this training/test split that including `rain` in the model instead of `rain.wetm` is slightly better.

Random training/test splits give different results!

Here are results for a few more random training/test splits:

```
> print( c(rss1,rss2) )
[1] 2.522239 2.521771
> print( c(rss1,rss2) )
[1] 2.416990 2.464476
> print( c(rss1,rss2) )
[1] 2.453752 2.500002
```

Clearly the **test/training split matters** – it is a random split which introduces randomness to results. This could be handled by repeating many (e.g. 50) times and averaging results (and reporting standard errors too). The process of repeating for different test/training splits is known as **cross-validation**.

K-fold cross-validation

A special case of cross-validation is when you split the data into K groups (typically $K = 5$, so 5-fold cross-validation, or $K = 10$) and fit K models – using each group as the test data once. Results tend to be less noisy than just using one training/test split (because each observation is used as a test observation once).

```
> library(DAAG)
> ft1 = lm(log(height)~temp+rain, dat=datheight)
> ft2 = lm(log(height)~temp+rain.wetm, dat=datheight)
> cv1 = cv.lm(data =datheight, ft1, m=5, plotit = FALSE) # 5 fold cross-validation
> cv2 = cv.lm(data =datheight, ft2, m=5, plotit = FALSE) # 5 fold cross-validation
> print( c( attr(cv1,"ms"),attr(cv2,"ms") ) )
[1] 2.31 2.35
```

5-fold cross validation suggests that the model with `rain` predicts slightly better than `rain.wetm`.

But recall that this was based on random splits – different splits will get different answers.

Trying again with different random splits (controlled through the seed argument):

```
> cv1 = cv.lm(data =datheight, ft1, m=5, seed=1, plotit = FALSE)
> cv2 = cv.lm(data =datheight, ft2, m=5, seed=1, plotit = FALSE)
> print( c(attr(cv1,"ms"),attr(cv2,"ms")) )
[1] 2.32 2.35

> cv1 = cv.lm(data =datheight, ft1, m=5, seed=50, plotit = FALSE)
> cv2 = cv.lm(data =datheight, ft2, m=5, seed=50, plotit = FALSE)
> print( c(attr(cv1,"ms"),attr(cv2,"ms")) )
[1] 2.38 2.45

> cv1 = cv.lm(data =datheight, ft1, m=5, seed=100, plotit = FALSE)
> cv2 = cv.lm(data =datheight, ft2, m=5, seed=100, plotit = FALSE)
> print( c(attr(cv1,"ms"),attr(cv2,"ms")) )
[1] 2.39 2.41
```

Results are looking pretty consistent!

How do you choose K ?

There is no single correct answer, so feel free to experiment.

You could use the $n^{3/4}$ rule again, although no-one ever with choosing the number of folds K . Instead it is common to use:

- N -fold or “leave-one-out” cross-validation for small datasets
- 10-fold CV for medium sized datasets (e.g. $n < 100$)
- 5-fold CV for large datasets

This is done for convention more than anything else, especially 10-fold. Again, general advice comes down to considering what your goal is for model selection, as well as experimenting to see how results vary with different K .

Information criteria

Another way to do model selection is to use the whole dataset (no training/test split) but to penalise more complex models in some way. The two most common ways:

$$\begin{aligned} AIC &= n \log \hat{\sigma}^2 + 2p \\ BIC &= n \log \hat{\sigma}^2 + \log(n)p \end{aligned}$$

(sometimes plus a constant), where p is the number of parameters in the model, n is sample size, and $\hat{\sigma}^2$ is the estimated error variance.

AIC – Akaike Information Criterion

BIC – Bayesian Information Criterion

The aim of the game is to

choose the model that minimises the information criterion

Where do these criteria come from?

AIC, like cross-validation, aims to minimise predictive error on new observations – but rather than estimating it directly, it uses approximate arguments from large sample theory.

Widely used, good for prediction, but overfits (*i.e.* models often larger than they should be).

BIC, as the name suggests, has a somewhat Bayesian motivation. Ironically, it is used mostly in non-Bayesian analysis! It was derived to maximise the “marginal likelihood” – choose the model that making the data as likely as possible.

Fairly widely used, good for variable selection (*i.e.* doesn’t overfit in large samples), but can underfit in small samples.

Computing IC's on R

Use the AIC or BIC function:

```
> ft.height1 = lm(log(height)~temp+rain, dat=datheight)
> ft.height2 = lm(log(height)~temp+rain.wetm, dat=datheight)
> c( AIC(ft.height1), AIC(ft.height2) )
[1] 658 660
> c( BIC(ft.height1), BIC(ft.height2) )
[1] 671 673
```

As before, these suggest the model with `rain` is a slightly better fit.

Pros and cons of information criteria

Information criteria have the advantage that there are no random splits in the data – you get the same answer every time. It is therefore simpler to interpret.

The disadvantages are that they are a little less intuitive and their validity relies on model assumptions (essentially, the fitted models need to be close to the correct model). Cross-validation requires only the assumption that the test/training data are independent – so it can still be used validly when you don't have a lot of confidence in the fitted model.

Ways to do subset selection

It's all well and good if you only have a few candidate models to compare, but what if you have a set of p predictor variables and you just want to find the subset that is best for predicting/explaining y ?

The common approaches:

- Forward selection – add one variable at a time, adding the best-fitting variable at each step
- Backward selection – add all variables then delete one variable at a time, deleting the worst-fitting variable at each step
- All-subsets – search all possible 2^p combinations. Not easy unless there are only a few variables (p small).

There are also hybrid approaches, such as stepwise selection, that do a bit of everything.

Stepwise selection on R

```
> ft.height = lm(log(height)~temp+rain+rain.wetm+temp.seas, dat=datheight)
```

```
> library(MASS)
```

```
> step <- stepAIC(ft.height)
```

```
> step$anova
```

Stepwise Model Path

Analysis of Deviance Table

Initial Model:

```
log(height) ~ temp + rain + rain.wetm + temp.seas
```

Final Model:

```
log(height) ~ temp + rain
```

	Step	Df	Deviance	Resid.	Df	Resid.	Dev	AIC
1					173		398	153
2	- rain.wetm	1	0.182		174		398	151
3	- temp.seas	1	3.585		175		401	151

If specifically wanting forward selection specify `direction="forward"`, etc.

All-subsets selection on R

```
> library(leaps)
> fit.heightallsub=regsubsets(log(height)~temp+rain+rain.wetm+temp.seas,
  data=datheight, nbest=2)
> summary(fit.heightallsub)
Subset selection object
...
Selection Algorithm: exhaustive
      temp  rain  rain.wetm  temp.seas
1 ( 1 )  "*"   " "    " "      " "
1 ( 2 )  " "   " "    "*"     " "
2 ( 1 )  "*"   "*"   " "      " "
2 ( 2 )  "*"   " "    "*"     " "
3 ( 1 )  "*"   "*"   " "      "*"
3 ( 2 )  "*"   "*"   "*"     " "
4 ( 1 )  "*"   "*"   "*"     "*"
> plot(fit.heightallsub)
```

Which method is best?

Again, there is no simple answer.

- All-subsets is more comprehensive, but not necessarily better and is computationally intensive.
- On the other hand, both forward and backward selection is not a good idea when you have many x variables (because you don't really want to use all of them)
- Multi-collinearity can seriously muck up subset selection

Penalised estimation

A modern and clever way to do subset selection is to use penalised estimation. Instead of estimating model parameters (β) to minimise least squares:

$$\min\left\{\sum_{i=1}^n (y_i - \mu_i)^2\right\}$$

we add a penalty as well which encourages estimates towards zero:

$$\min\left\{\sum_{i=1}^n (y_i - \mu_i)^2 + \lambda \sum_j |\beta_j|\right\}$$

This is known as the LASSO (Least Absolute Shrinkage and Subset selection Operator, Tibshirani, 1996). It is used for example in MAXENT software.

The LASSO can equivalently be thought of as constrained minimisation:

$$\min\left\{\sum_{i=1}^n (y_i - \mu_i)^2\right\} \text{ such that } \sum_j |\beta_j| \leq t$$

Penalty \Rightarrow less variance, more bias

Penalising introduces bias: the bad news is that pushing estimates of β towards zero makes most values closer to zero than they should be.

Penalising reduces variance: the good news is that pushing estimates of β towards zero reduces their standard error.

Penalised estimation is a good thing to do when:

- the main goal is prediction – it tends to improve predictive performance (by reducing variance)
- you have lots of parameters in your model and/or not a large sample size – in such cases variance is an important issue, plus it's fast!

Nuisance parameter in penalty term

There is a “nuisance parameter” to estimate in LASSO: λ (or t). The value of this parameter determines how hard we push the slope parameters towards zero – i.e. how much we bias estimates, in an effort to reduce variance. **“bias-variance trade-off”**

λ is large \Rightarrow most $\beta_j = 0$

λ is small \Rightarrow few $\beta_j = 0$

The problem of estimating the nuisance parameter is pretty much the same as the problem of deciding how many terms to include in the model. So we can use the same approaches – cross-validation, BIC, and so on.

Basically, penalised estimation converts the problem of selecting from 2^p models to selecting one value of λ .

LASSO on R

```
> library(glmnet)
> X=cbind(datheight$temp, datheight$rain, datheight$rain.wetm, datheight$temp.seas)
> ft.heightcv=cv.glmnet(X, log(datheight$height))
> plot(ft.heightcv)
> ft.lasso=glmnet(X, log(datheight$height), lambda=ft.heightcv$lambda.min)
> ft.lasso$beta
```

Pros and cons of LASSO

Good news about the LASSO:

- It's fast
- It predicts well (by reducing variance)
- It simplifies the problem of model selection to one of estimating a single parameter (λ)

The bad news:

- It biases parameter estimates – relationships are flatter than they should be
- How to get standard errors?

Mixed effects models

- Random effects
- Linear mixed effects model
- Likelihood functions
- Inference about mixed effects models

Random effects

Do you have in your design:

- A factor which takes a large number of potential levels
- Only a random sample of these levels has actually been included in your study.

Then you have yourself a **random factor**. You can incorporate it into your model using **random effects**.

Mathematically, this puts a distribution on the coefficients for that factor.

Any factor that is not treated as random is referred to as **fixed**. To this point, we have treated everything as fixed (“**fixed effects** models”).

Example: Estuaries

Graeme is interested in the effects of water pollution on subtidal marine micro-invertebrates – in particular, what is the effect on invertebrate abundance? He samples in seven estuaries along the New South Wales coast (three of which are “Pristine”, four are “Modified”), and in each estuary, he takes 4-7 samples and counts the creepy crawlies therein.

What factors are there? Fixed or random?

Study design diagram:

This is an example of the most common scenario when a random effect pops up, **nested factors**:

Factor B is nested within A if each level of B only occurs within one level of A .

Graeme's estuaries were nested within modification – each of the seven estuary (B) was classified as one of “modified” and “pristine” (A).

Nested factors are not necessarily random, but they should be – otherwise you would have a tough job making inferences about factor A .

Another place random effects commonly pop up is in hierarchical or “multi-level” designs – when there are multiple levels at which sampling is done (region, site, species, ...)

“Pseudo-replication” – where one takes replicates at a level not of primary interest – can be considered as a multi-level design.

Fitting models with random effects

We will use the R package `lme4`. Doug Bates (the author of the package) has been writing a whole text on it, available at: <http://lme4.r-forge.r-project.org/IMMwR/lrgprt.pdf>

In the case of normal responses, models are fitted using the `lmer` function, with random effects specified in brackets.

```
> datSmall = read.csv("estuarySmall.csv", header = T)
> library(lme4)
> ft.estu = lmer(Total~Mod+(1|Estuary), data=datSmall)
> summary(ft.estu)
Linear mixed model fit by REML [lmerMod]
Formula: Total ~ Mod + (1 | Estuary)
Data: datSmall
```

REML criterion at convergence: 314

...

Random effects:

Groups	Name	Variance	Std.Dev.
Estuary	(Intercept)	10.7	3.27
	Residual	123.7	11.12

Number of obs: 42, groups: Estuary, 7

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	39.05	3.24	12.07
ModPristine	-11.24	4.29	-2.62

Is there any effect of modification on invertebrate abundance?

(Huh|What)?

In R formulas, 1 means fit a y -intercept ($1 \times \beta_0$). Using $(1|\text{Estuary})$ means shift the y -intercept to a different value for each level of the factor Estuary. The symbol “|” means “given” or “conditional on” – so we are saying that the value of the y -intercept depends on Estuary, it needs to be different for different levels of Estuary.

In short, this introduces a random intercept for Estuary.

You can use this notation to introduce random slopes also – $(\text{pH}|\text{Estuary})$ would fit a different slope against pH in each estuary.

Linear mixed effects model

The linear mixed effects model is as follows:

$$\begin{aligned}y_i &\sim \mathcal{N}(\mu_i, \sigma^2) \\ \mu_i &= \beta_0 + \mathbf{x}_i^T \boldsymbol{\beta} + \mathbf{z}_i^T \mathbf{b} \\ \mathbf{b} &\sim \mathcal{N}(0, \Sigma) \text{ independently of } y_i\end{aligned}$$

Basically, it looks just like the standard (fixed effects) linear model, except for the third line – this line says that some of the coefficients in the model are random, normally distributed, and independent of y_i .

(Graeme had a model with random intercepts for Estuary, i.e. the \mathbf{z}_i are indicator variables for each Estuary.)

Linear mixed effects model assumptions

1. The observed y values are **independent** (after conditioning on x)
2. The y values are **normally distributed** with **constant variance**

$$y \sim \mathcal{N}(\mu_y, \sigma^2)$$

3. **straight line relationship** between mean of y and each x (and z)

$$\mu_i = \beta_0 + \mathbf{x}_i^T \boldsymbol{\beta} + \mathbf{z}_i^T \mathbf{b}$$

(As before, when x -variables are factors, this assumption doesn't matter)

4. The random effects \mathbf{b} are **independent of y .**
5. The random effects \mathbf{b} are **normally distributed**, sometimes with **constant variance**

Look familiar?

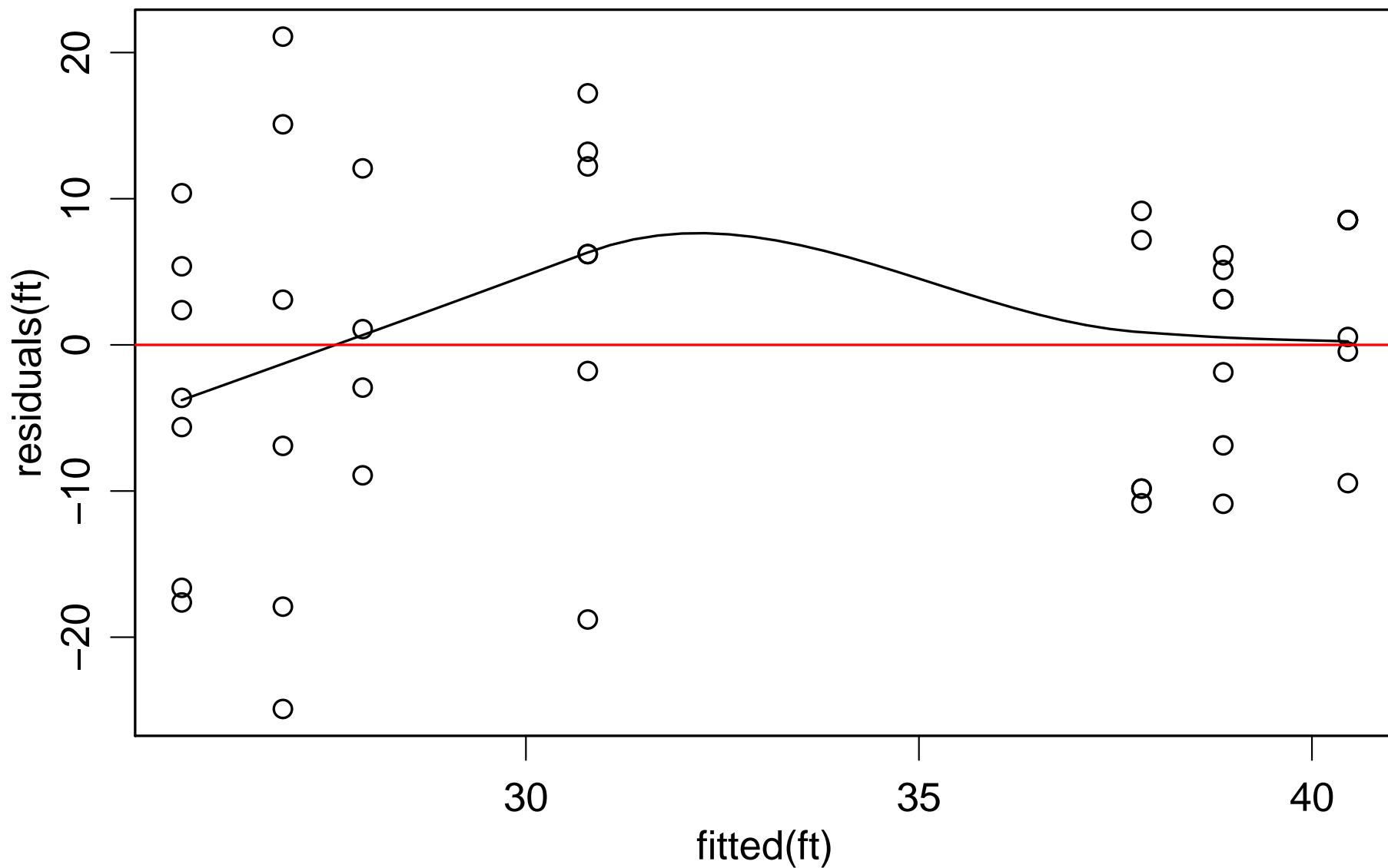
Importance of these assumptions

1. **independent y** can be guaranteed to be satisfied – **how???**
2. **normality** of y doesn't really matter (due to Central Limit Theorem) except for small samples/strongly skewed data/outliers. Check on a normal quantile plot.
constant variance check a **residual plot** for no fan-shape
3. **Straight line relationship** is crucial if x is quantitative – Check for no pattern on **residual plot**, U-shape is bad news.
4. **b independent of y** can be guaranteed to be satisfied – **how???**
5. **normality** of b doesn't seem to matter much, **if** your research question is concerning the fixed effects β

```
> ft.estu = lmer(Total~Mod+(1|Estuary), data=datSmall)
> scatter.smooth(residuals(ft.estu)~fitted(ft.estu))
> abline(h=0, col="red")
```

Or to plot residuals against “unconditional” predicted values (using the fixed effects term only):

```
> scatter.smooth(residuals(ft.estu)~predict(ft.estu,REform=NA))
> abline(h=0, col="red")
```



So do the model assumptions appear satisfied?

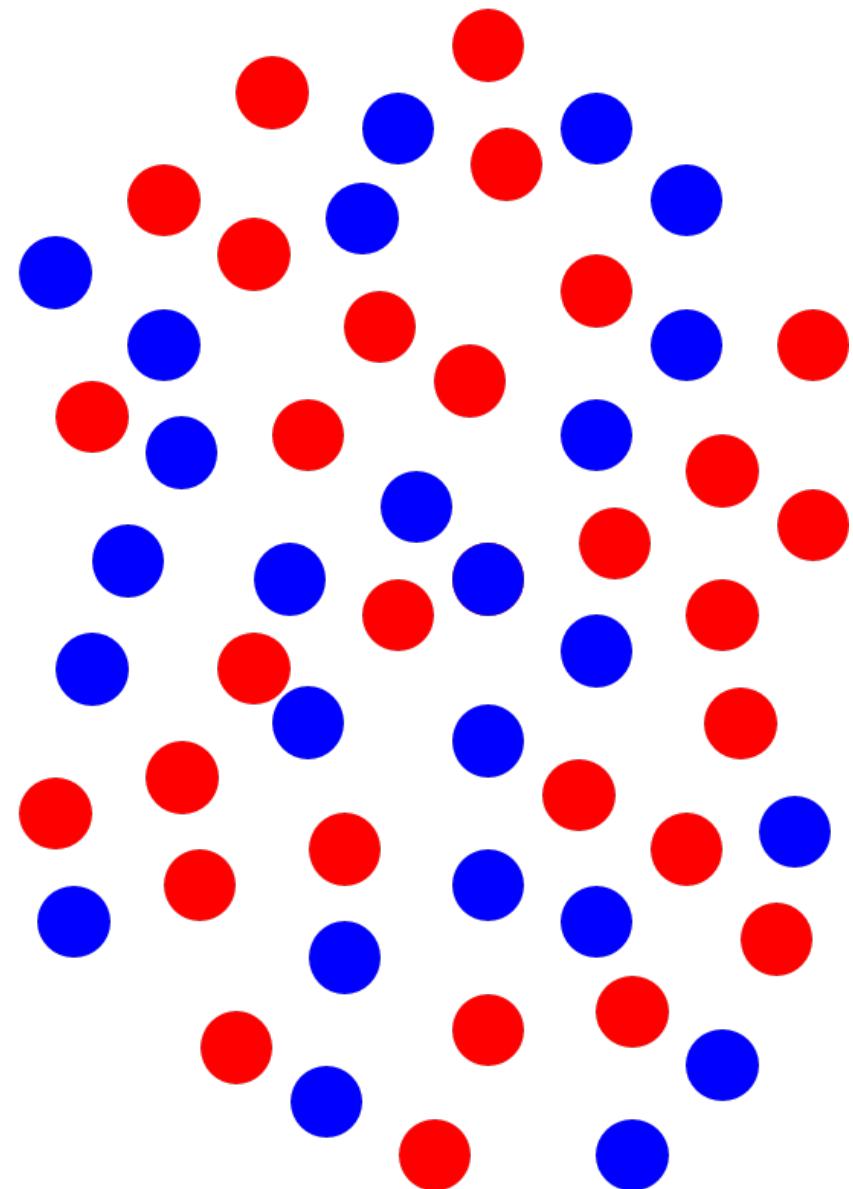
Likelihood functions

Recall that (fixed effects) linear models are fitted by **least squares** – minimise the sum of squared error in predicting y from x . Mixed effects models are fitted by **maximum likelihood** or **restricted maximum likelihood**.

The likelihood function, for a given value of model parameters, is the joint probability density of your data. The higher the value, the more likely your data.

The key idea of **maximum likelihood** estimation is to choose as your parameter estimates the values that make your data most likely, *i.e.* the values for parameters that would have given the highest probability of observing the values of the response variables you actually observed.

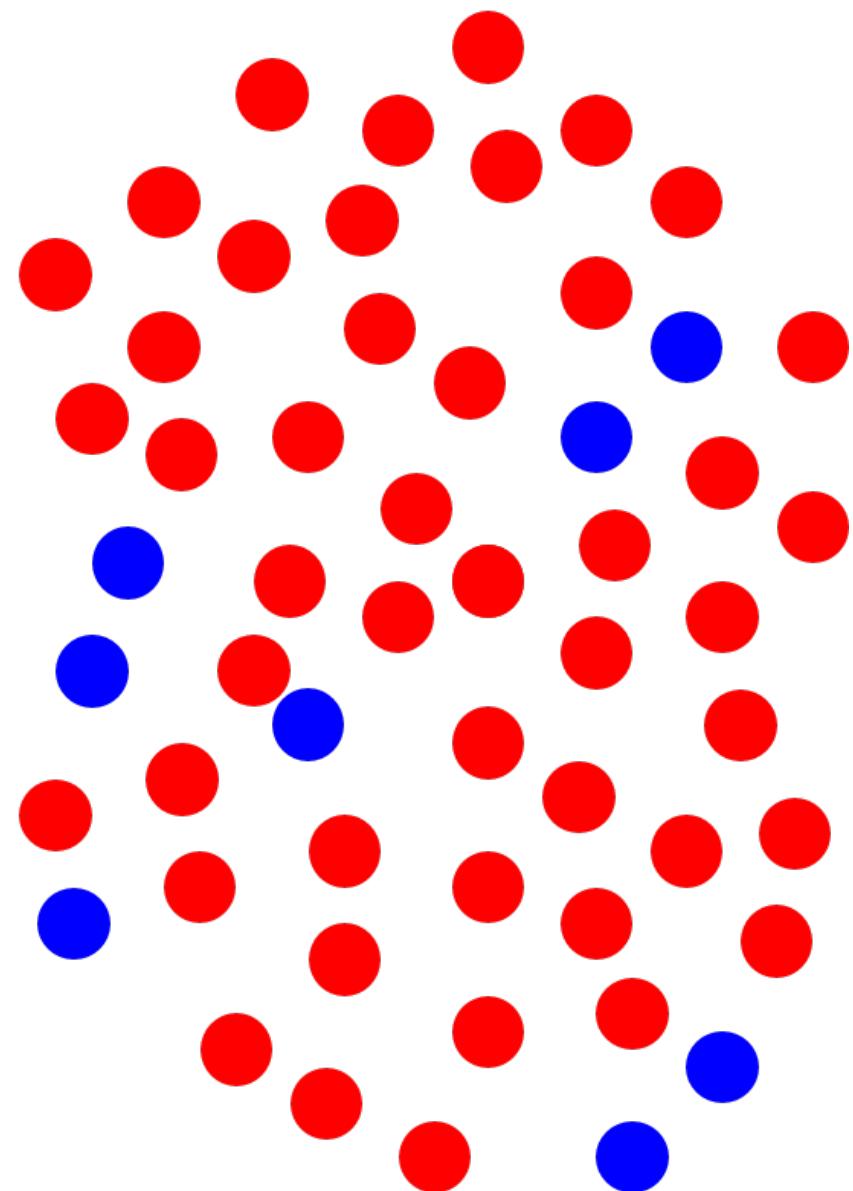
known sample

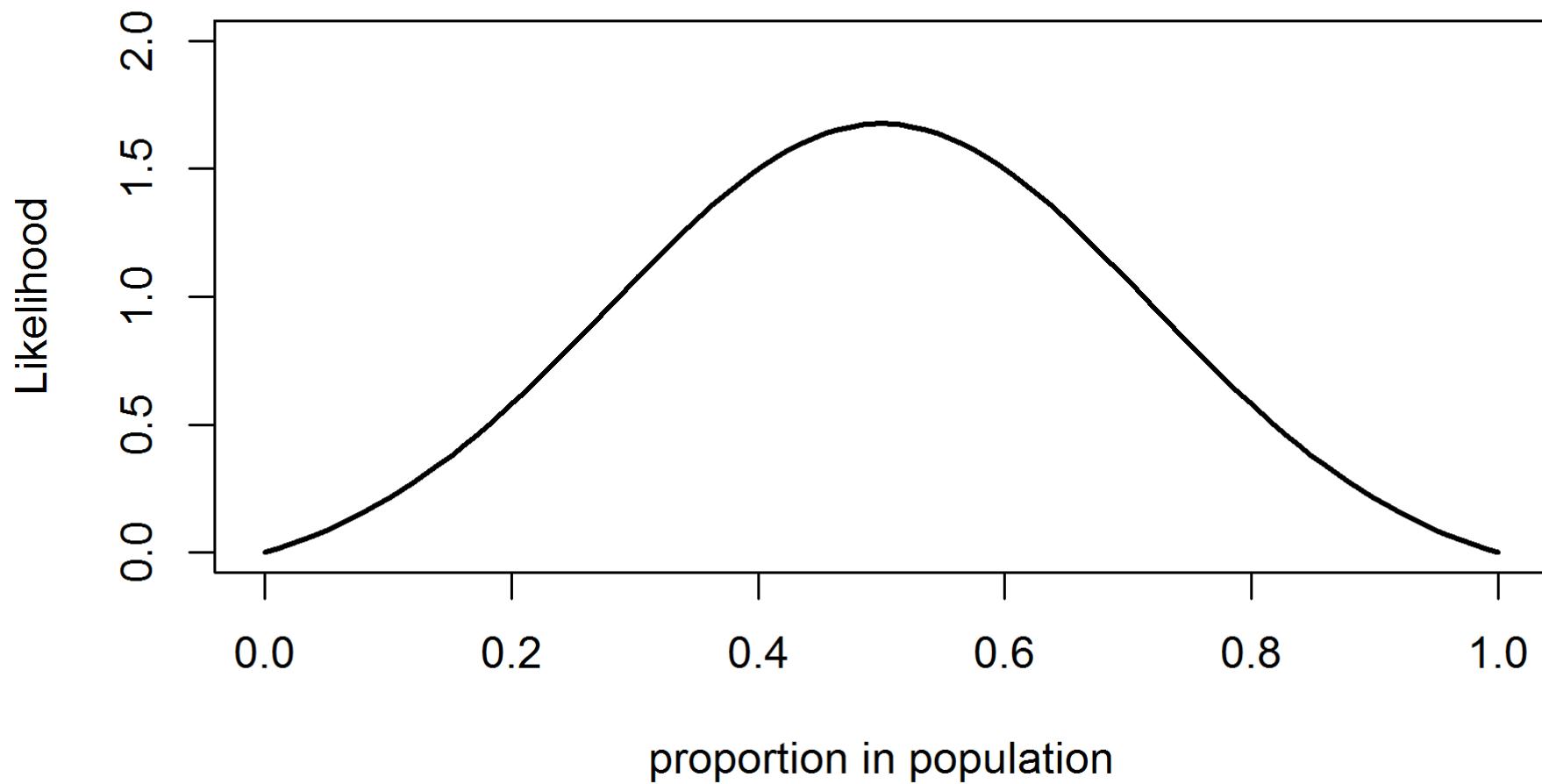


known sample



possible population





Properties of maximum likelihood estimation

Under a broad set of conditions which most models satisfy, maximum likelihood estimators:

- are consistent (in large samples, they go to the right answer)
- are asymptotically normal (which makes inference a lot easier)
- are efficient (they have minimum variance amongst consistent, asymptotically normal estimators)

Basically, they are awesome, and these properties give most statisticians license to base their whole world on maximum likelihood – provided that you can specify a plausible statistical model for your data (you need to know the right model so you are maximising the right likelihood).

Restricted maximum likelihood = least squares

Restricted maximum likelihood (REML) is a cheat fix to ensure all parameter estimates are exactly unbiased when sampling is balanced (they are only approximately unbiased for maximum likelihood, not exactly unbiased).

This is worth doing if your sample size is small relative to the number of terms in the model.

In `lme4`, all linear mixed models are fitted using REML by default.

Inference from mixed effects models

Inference from mixed effects models is a little complicated, because the likelihood theory which usually holds sometimes doesn't (in particular, asymptotic normality) when you have random effects.

Note in the following slide that there are no P -values for the random effects nor the fixed effects – these were deliberately left out because the authors of `lme4` are a little apologetic about them.

t values give you a rough idea though, anything larger than 2 is probably significant at the 0.05 level.

(Can also try the `nlme` package for slightly more friendly output.)

```
> ft.estu = lmer(Total~Mod+(1|Estuary), data=datSmall)
> summary(ft.estu)
```

Linear mixed model fit by REML
Formula: Total ~ Mod + (1 | Estuary)
Data: datSmall

Random effects:

Groups	Name	Variance	Std.Dev.
Estuary	(Intercept)	10.7	3.27
	Residual	123.7	11.12

Number of obs: 42, groups: Estuary, 7

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	39.05	3.24	12.07
ModPristine	-11.24	4.29	-2.62

Correlation of Fixed Effects:

(Intr)
ModPristine -0.755

Comparing mixed effects models

You can use the `anova` function as usual to compare models. This uses a **likelihood ratio test** (comparing the maximised likelihood under the null and alternative models).

However, it is advised that you use `REML=FALSE` when fitting the models to be compared (*i.e.* models should be fitted by standard maximum likelihood to do a likelihood ratio test). And take the results with a grain of salt, as again they are only approximate.

```

> ft.estu = lmer(Total~Mod+(1|Estuary), data=datSmall, REML=FALSE)
> ft.estuInt = lmer(Total~(1|Estuary), data=datSmall, REML=FALSE)
> anova(ft.estuInt, ft.estu)

Data: datSmall

Models:
ft.estuInt: Total ~ (1 | Estuary)
ft.estu: Total ~ Mod + (1 | Estuary)

      Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)
ft.estuInt  3 334 339   -164       328
ft.estu      4 330 337   -161       322  6.04      1     0.014 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

```

Is there an effect of modification?

You can use `anova` to similarly test for random effects, but this gets a little complicated:

- It doesn't work when you have a single random effect (as Graeme does), Zuur *et al* (2009) proposes a workaround using the `nlme` package and `gls`.
- The P -values are approximately double what they should be, but even then are very approximate.

We shall focus on this problem in the next session...

Confidence intervals for parameters

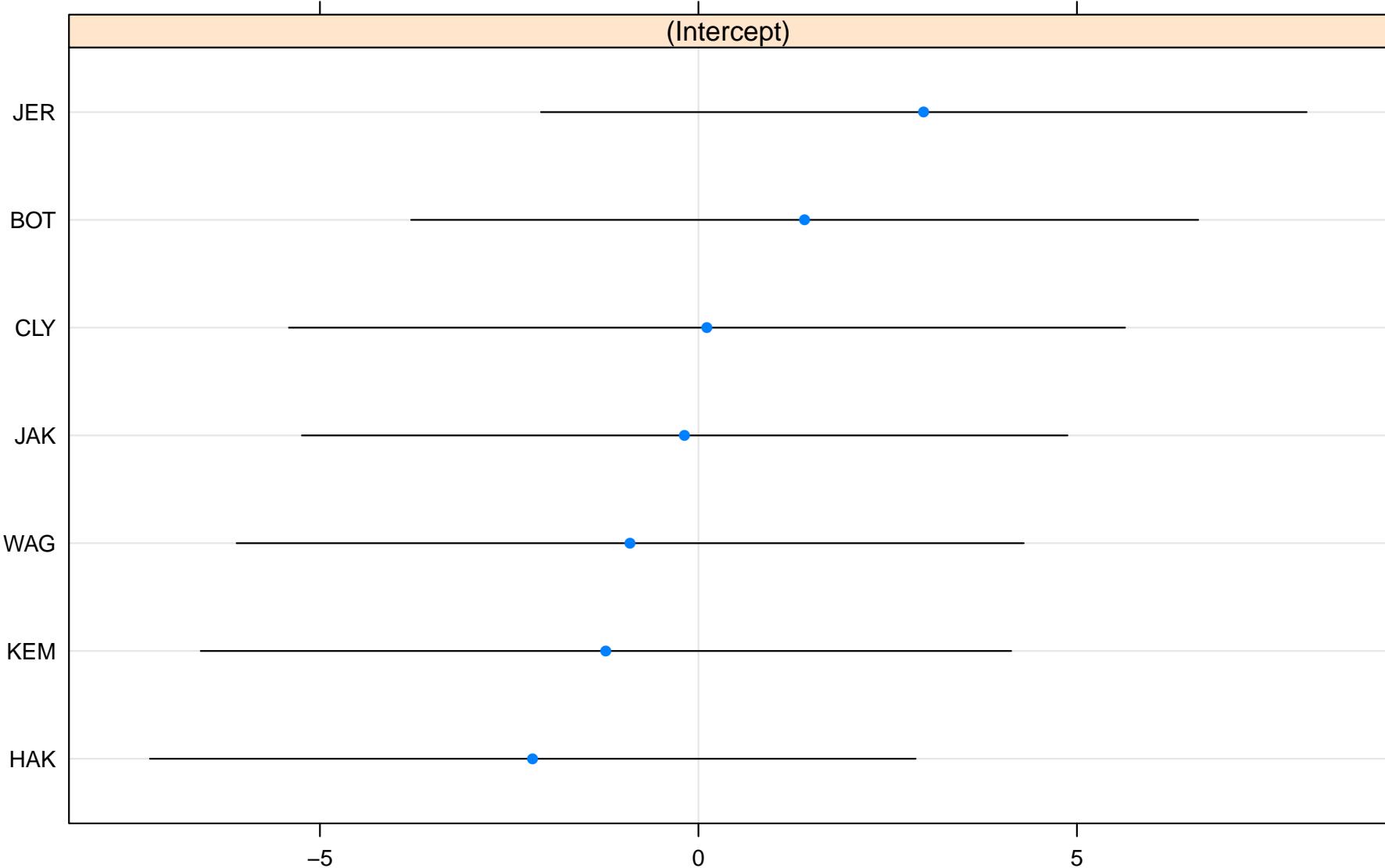
Use the `confint` function (just like for `lm`):

```
> confint(ft.estu)
Computing profile confidence intervals ...
      2.5 % 97.5 %
.sig01      0.00   7.61
.sigma      8.94  14.08
(Intercept) 32.82  45.25
ModPristine -19.46 -2.99
```

For random effects the mechanics of inference are messier, but try:

```
> rft=ranef(ft.estu, condVar=TRUE)
> dotplot(rft)
```

Interpret these ranges more-or-less as 95% confidence intervals for the random effects (true values of shift in each Estuary).



Is there any evidence of an effect of estuary?

Mixed effects models (cont'd)

- What if I want more accurate inferences?
- Other issues in mixed effects modelling
- Correlated random effects: for correlation in space or time
- Generalised linear mixed models (GLMMs)

What if I want more accurate inferences?

This can be done but it requires a bit of coding (and a lot of computation time – especially for large datasets). The best of the immediate options is the parametric bootstrap:

- using the `simulate` or `bootMer` function and writing your own parametric bootstrap
- for hypothesis testing two mixed models, the `PBmodcomp` from the `pbkrtest` package
- for confidence intervals, the `confint` with `method = “boot”`
- for linear, generalized linear and mixed models the `bootLRT` and `bootSE` functions available from Moodle in the file `boot.R`

Example parametric bootstrap code, to test for effect of Estuary using the bootLRT function. The boot.R file will need to be in the working directory.

```
> source("boot.R")
> datSmall = read.csv("estuarySmall.csv", header = T)
> datSmall$AmphiPA <- as.numeric(datSmall$Amphipod.tubes>0)
> ft.noestu = lm(Total~Mod, data=datSmall)
> ft.yesestu = lmer(Total~Mod+(1|Estuary), data=datSmall, REML = FALSE)
> bootLRT(ft.noestu,ft.yesestu,1000)
[1] "Model 1"
Total ~ Mod
[1] "Model 2"
Total ~ Mod + (1 | Estuary)
[1] "1000 simulations"
[1] "Bootstrap p-value = 0.178"
```

Is there any evidence of an effect of estuary?

Example parametric bootstrap code, to test for effect of modification using the PBmodcomp function.

```
> library(pbkrtest)
> ft.nomod = lmer(Total~(1|Estuary), data=datSmall, REML = FALSE)
> ft.mod = lmer(Total~Mod+(1|Estuary), data=datSmall, REML = FALSE)
> #larger model first
> PBmodcomp(ft.mod,ft.nomod)

Parametric bootstrap test; time: 23.00 sec; samples: 1000 extremes: 24;
large : Total ~ Mod + (1 | Estuary)
small : Total ~ (1 | Estuary)
stat df p.value
LRT    6.04   1   0.014 *
PBtest 6.04       0.025 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1    1
```

Is there any evidence of an effect of modification?

Same test for modification using the bootLRT function.

```
#smaller model first  
> bootLRT(ft.nomod,ft.mod,1000)  
[1] "Model 1"  
Total ~ (1 | Estuary)  
[1] "Model 2"  
Total ~ Mod + (1 | Estuary)  
[1] "1000 simulations"  
[1] "Bootstrap p-value = 0.033"
```

Any difference between bootLRT and PBmodcomp?

Parametric bootstrap standard error of fixed effects using bootSE function:

```
> ft.yesestu = lmer(Total~Mod+(1|Estuary), > data=datSmall, REML = FALSE)
summary(ft.yesestu)
```

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	39.09	2.79	14.03
ModPristine	-11.27	3.69	-3.06

```
> bootSE(ft.yesestu,1000)
```

paramater	Std	Covariance
1 (Intercept)	2.77	7.65 -7.45
2 ModPristine	3.72	-7.45 13.81

How does this compare to the standard error from the original model fit?

Other random effects issues

What about unbalanced sampling?

There are some texts (usually in their tenth edition) which advise that mixed effects models require sampling to be balanced, *i.e.* the same number of replicates for all observations of each treatment group.

This is not the case.

It is a great idea to have balanced sampling – better for power and robustness to assumption violations (especially equal variance assumption). But it is not necessary.

Old methods of fitting mixed effects models (via sums of squares decompositions) required balanced sampling for random effects estimation. (Restricted) maximum likelihood estimation however has no such constraint.

On the topic of sample size...

When you have random effects, there are now multiple sample sizes to worry about.

n – total sample size. The larger it is, the better your estimates of lower-level fixed effects (and residual variance).

n_B – the number of levels of factor B . The larger it is, the better your estimate of the effect of B , and any higher-level effects depending on B .

You could have a million observations but if you only have a few levels of your random effect you have very little information about B and hence about anything B is nested in.

e.g. If Graeme had taken 100 samples at each of four estuaries he would probably have less of an idea about the effect of modification than he does now, with his 4-7 samples at each of seven estuaries.

Random factors don't have to be random effects

Just because a factor is random doesn't mean that it has to be treated as a random effect in modelling.

Use random effects if **both** the following conditions are satisfied:

- If you have a random factor (*i.e.* large number of levels, from which you have a random sample)
- You want to make general inferences with respect to all possible levels of the random effect, not just those that were sampled.

If you are happy making inferences conditional on your observed set of levels of the random factor then there is no harm in treating the effect as fixed and saving yourself some pain and suffering (although this not always possible if n_B is large!)

Random vs fixed: pros and cons

Fixed effects are good because they are easier for estimation and for inference. Random effects are good because **inferences at higher levels in the hierarchy are still permissible even when there is significant variation at lower levels.**

For example, if Estuary was treated as a fixed effect, then if different estuaries have different invertebrate abundances, we cannot make any inferences about the effect of modification Mod. If we know that different estuaries have different abundance, then by default Mod will have different abundance (because different estuaries have different levels of Mod).

Put another way, a fixed effect of Estuary will be completely confounded with the effect of modification

But if Estuary is treated as a random effect, we can estimate the variation in abundance due to estuary and ask if there is a mod effect above and beyond that due to variation with estuary.

What if my factor isn't really random?

Recall that for a factor to be random, the levels of it used in your study need to be a **random sample** from the population of possible values.

What if I need to treat my factor as random (to make inferences about higher-level effects) but I didn't actually sample levels of this factor at random?

Well that's a bit naughty =D

Wherever possible, if you want to treat a factor as random in analysis, you should sample the levels of the factor randomly. Amongst other things, this ensures independence assumptions are satisfied, and that you can make valid inferences about higher-level terms in the model.

Other uses of random effects

Having said that, the interpretation of random effects as randomly chosen levels is stretched a bit; random effects are frequently used as a mathematical device which can:

- **Induce correlation between groups of correlated observations** (e.g. invertebrate abundance across samples from the same estuary are correlated due to biotic interactions). This idea can be extended to handle spatial or temporal correlation.
- **Stabilise parameter estimates** when there are lots of parameters (fixed effects models work well when the number of parameters is small compared to the sample size n).

In these instances, we are using random effects as a mathematical device rather than using them to reflect (and generalise from) study design. A cost of doing this is that inference becomes more difficult.

Correlated random effects: for correlation in space or time

Sometimes you have correlation between observations which you believe has a particular type of structure. Some major examples:

- **Repeated measures in time** – often you can assume the correlation decreases as a (possibly known) function of time between measurements
- **Spatial autocorrelation** – often observations are correlated spatially, the farther apart they are, the weaker the correlation
- **Phylogeny** – often responses are correlated across taxa, the more closely related the species are, the greater the correlation

Random effects can be used to handle these situations, by introducing correlated random effects. The `nlme` package is good for observations that are correlated in space or time.

Some introductory examples for temporal and spatial autocorrelation can be found on the Eco-Stats blog
(follow link from our homepage, www.eco-stats.unsw.edu.au).

Penalised estimation as a mixed effects model

Recall the LASSO – a shrinkage method for improving predictive performance by reducing the variance in parameter estimates. The LASSO minimises:

$$\min \left\{ \sum_{i=1}^n (y_i - \mu_i)^2 + \lambda \sum_j |\beta_j| \right\}$$

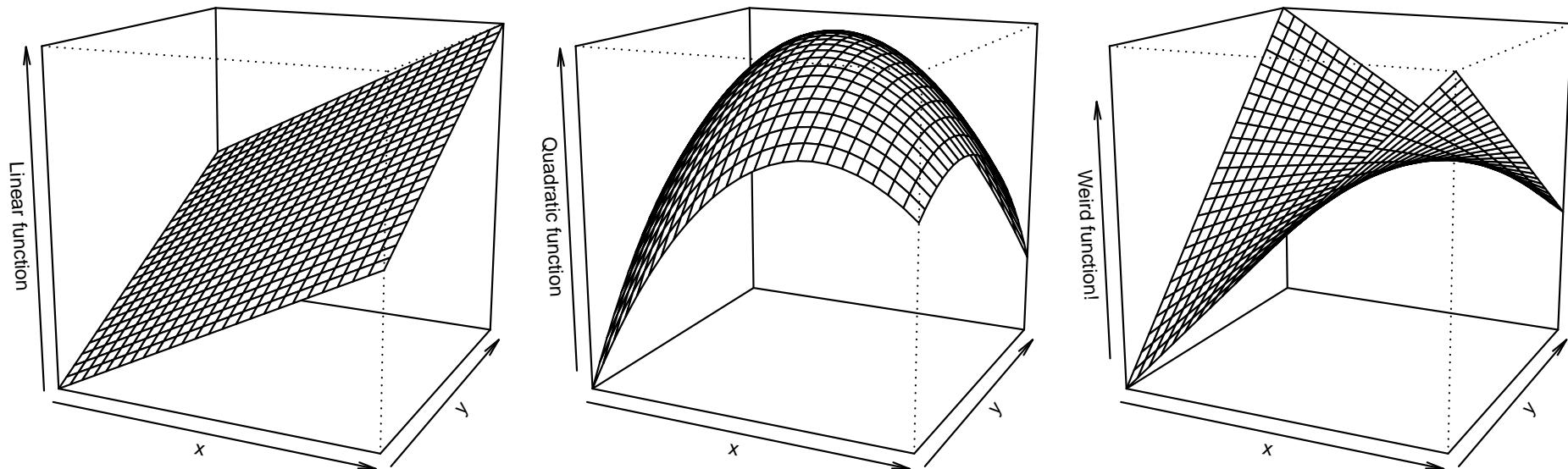
The LASSO can be thought of as putting random effects on regression parameters – but not assuming they are normal, instead assuming they come from a longer-tailed distribution with a big peak at zero (“double exponential”).

Hence the LASSO is an example of using random effects as a mathematical device – to stabilise parameter estimates.

Wiggly Models

- Spline smoothers
- Smoothers with interactions
- Smoothers as diagnostic tools
- Cyclical variables

Recall that a “linear model” does not need to be linear: by including functions of x as predictors (e.g. quadratic or cubic terms), you can use the linear model function to fit some non-linear functions:



There are lots of other cool functions you can fit too in the linear modelling framework. We will look at two:

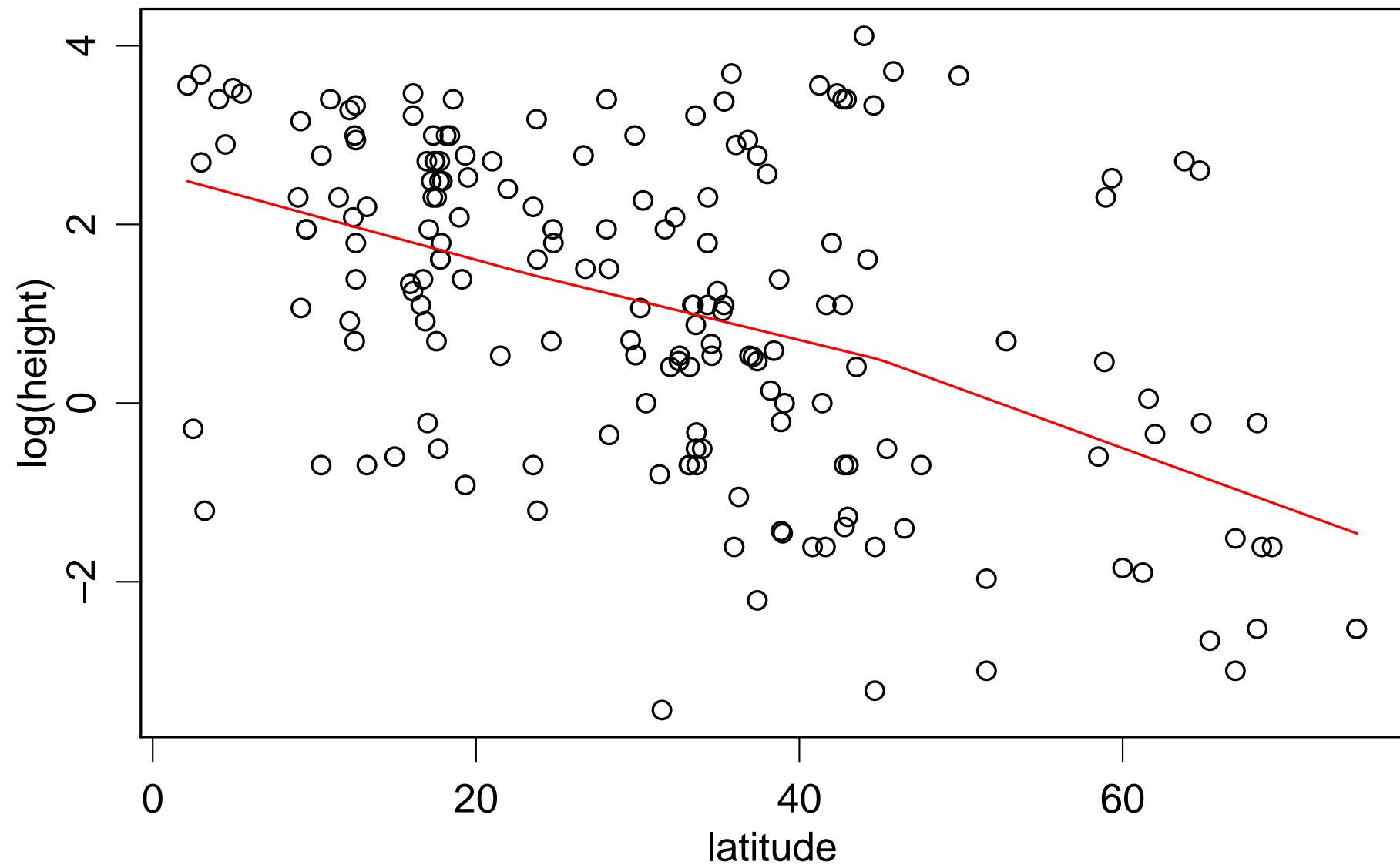
- spline smoothers
- periodic functions

Spline smoothers

Spline smoothers break your explanatory variable into bits (at **knots**) and fits a multiple regression against all of these bits (with a penalty for smoothness).

A simple example of this is piecewise linear model fits (used in MAX-ENT software).

A piece-wise linear fit (with changes of slope at 23 and 45 degrees)

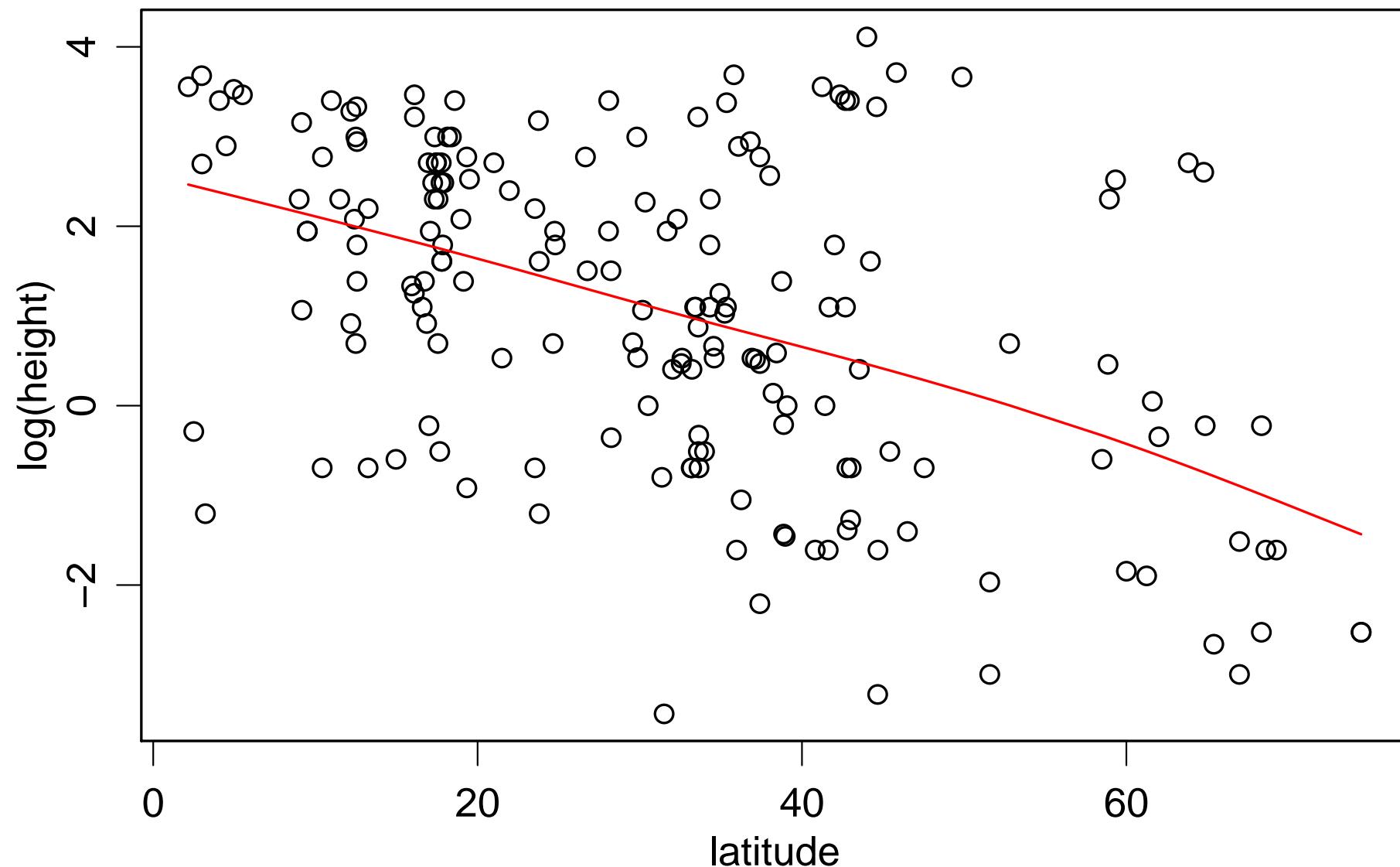


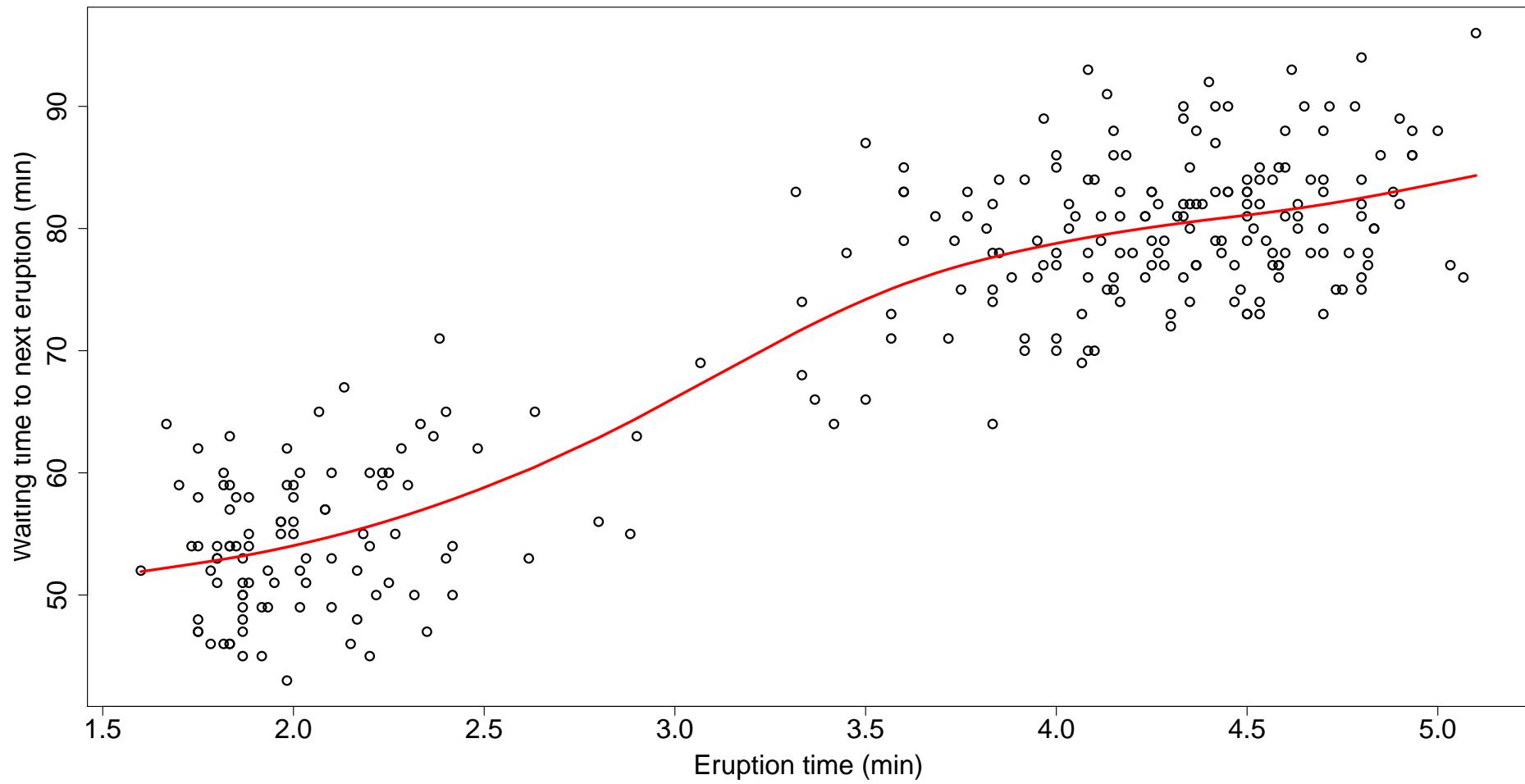
Piecewise linear fits are a bit old school (at least for functions of one variable). They don't look smooth and in most problems a "kinky" function is not realistic (e.g. why suddenly change slope at 45 degrees?).

A more common approach is to keep the function smooth by fitting cubics (or similar) where only the cubic term is broken into pieces – linear and quadratic terms are not.

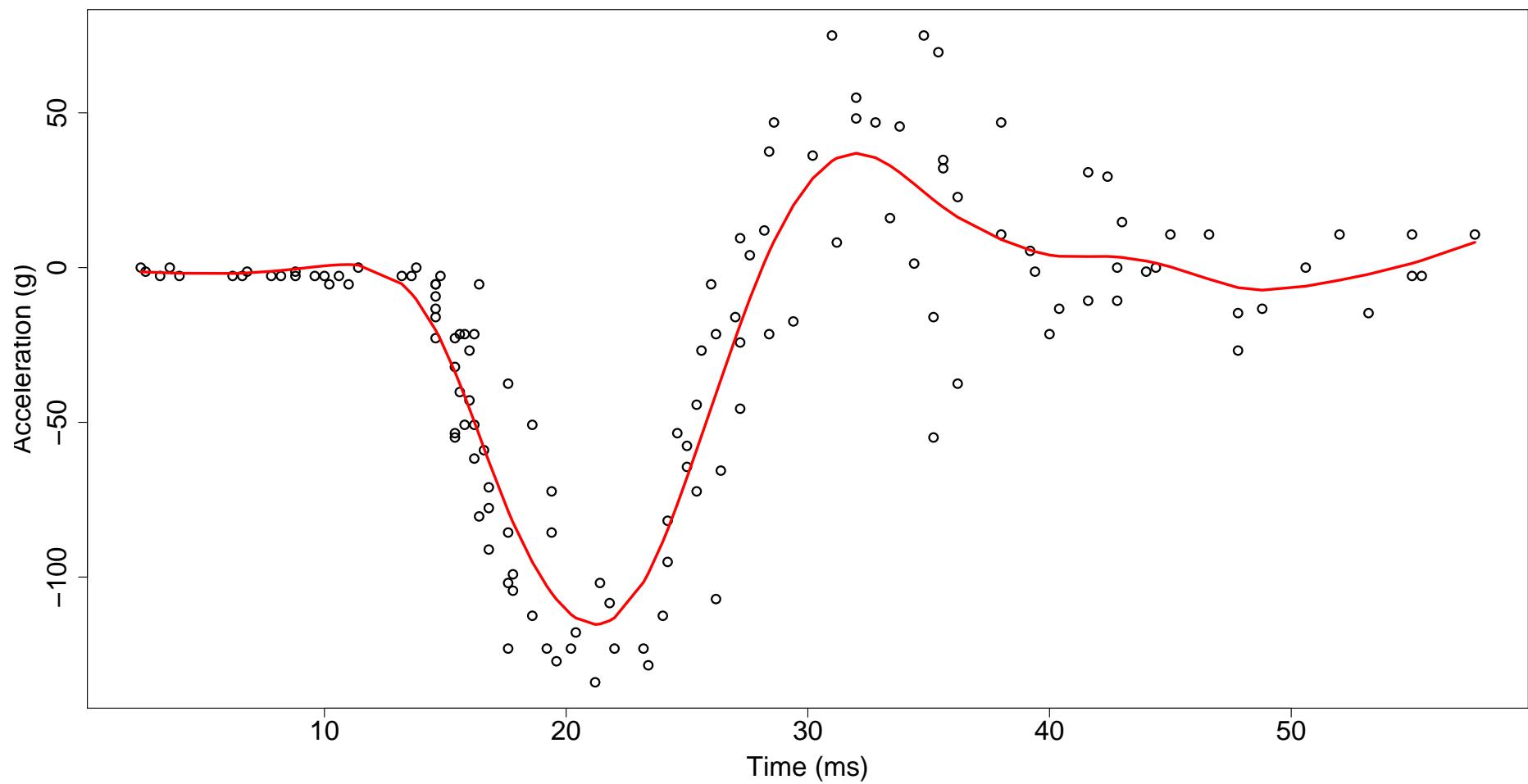
These function are known as **Spline smoothers**.

Spline smoothers fit a smooth curve to data





9.8

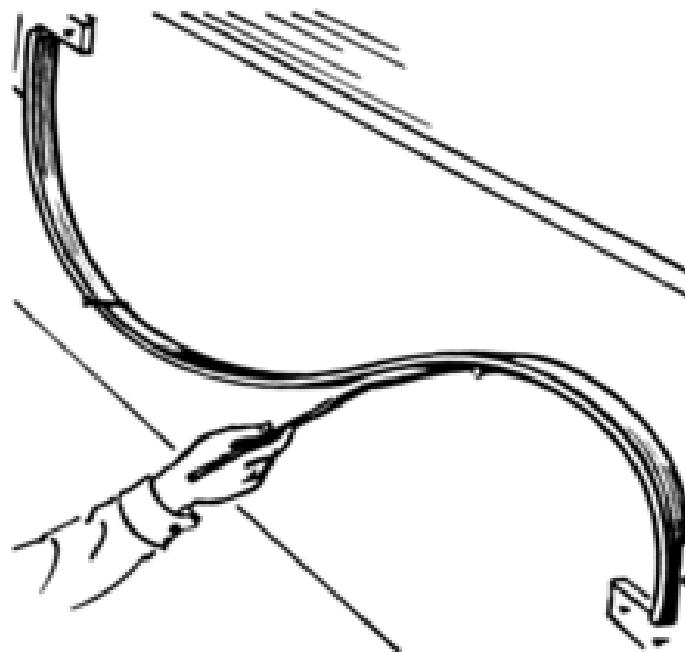


9.9

Why “splines”?

The term spline comes from woodworking (especially shipbuilding), and the problem of bending straight sections of wood to form curves (e.g. ship hull).

This is done by fixing at control points or “knots” .



The statistician who came up with the term “spline smoothing” clearly spends a lot of time in their garage...

How to fit on R

First load the `mgcv` package.

Then use `s(lat)` in your formula (for a spline for latitude. `s` for spline).

```
> library(mgcv)
> datheight <- read.csv("plantHeightSingleSpp.csv")
> ft.heightgam = gam(log(height)~s(lat),dat=datheight)
> summary(ft.heightgam)
```

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.055	0.119	8.85	9.1e-16 ***

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Approximate significance of smooth terms:

edf	Ref.df	F	p-value
s(lat)	1	1	53.4 6e-12 ***

R-sq.(adj) = 0.229 Deviance explained = 23.3%

GCV = 2.5594 Scale est. = 2.5307 n = 178

Any evidence of an effect of latitude?

Number of knots

You can specify an upper limit to the number of knots as `k`, an argument to the spline:

```
> ft.heightgam = gam(log(height)~s(lat,k=5),dat=datheight)
```

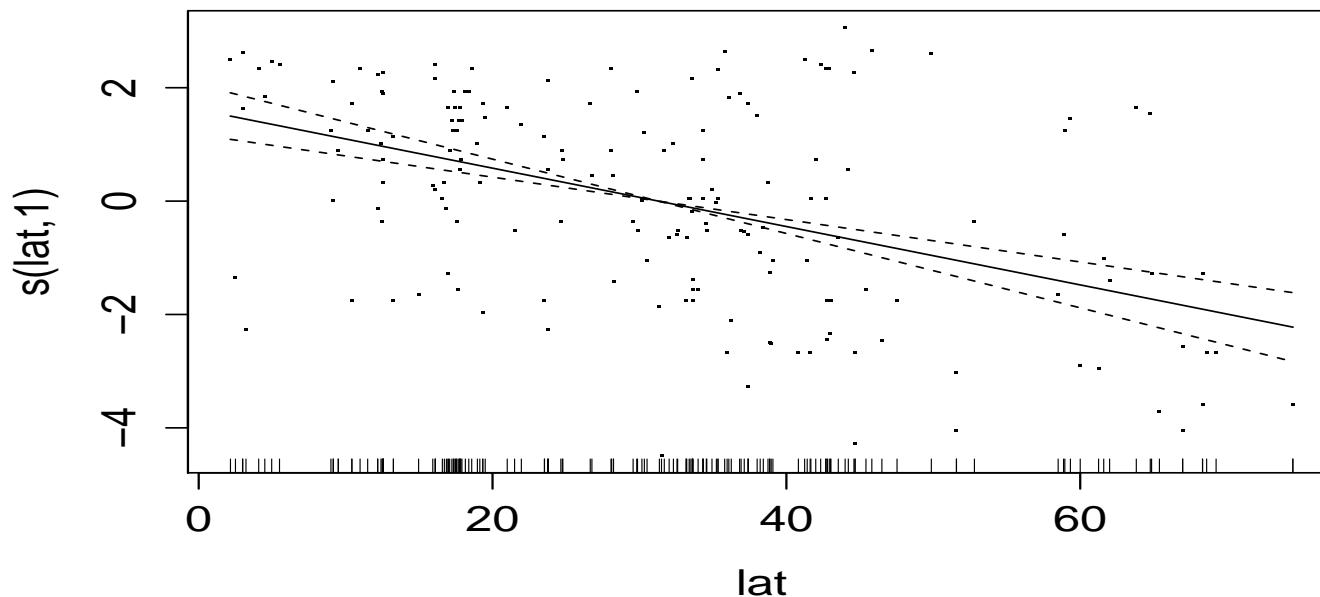
By default `k` is usually somewhere near 10 (but it depends).

The more knots, the more wiggly your smoother can be, but it takes longer to fit. If you think your fit might need to be extra wiggly it is a good idea to try changing `k` to a larger value (20? 50?) to see if it changes much.

Diagnostics on R using `gam()`

You have to do a little work to get a residual plot. The `plot` function does not plot residuals, it plots the fitted smoother:

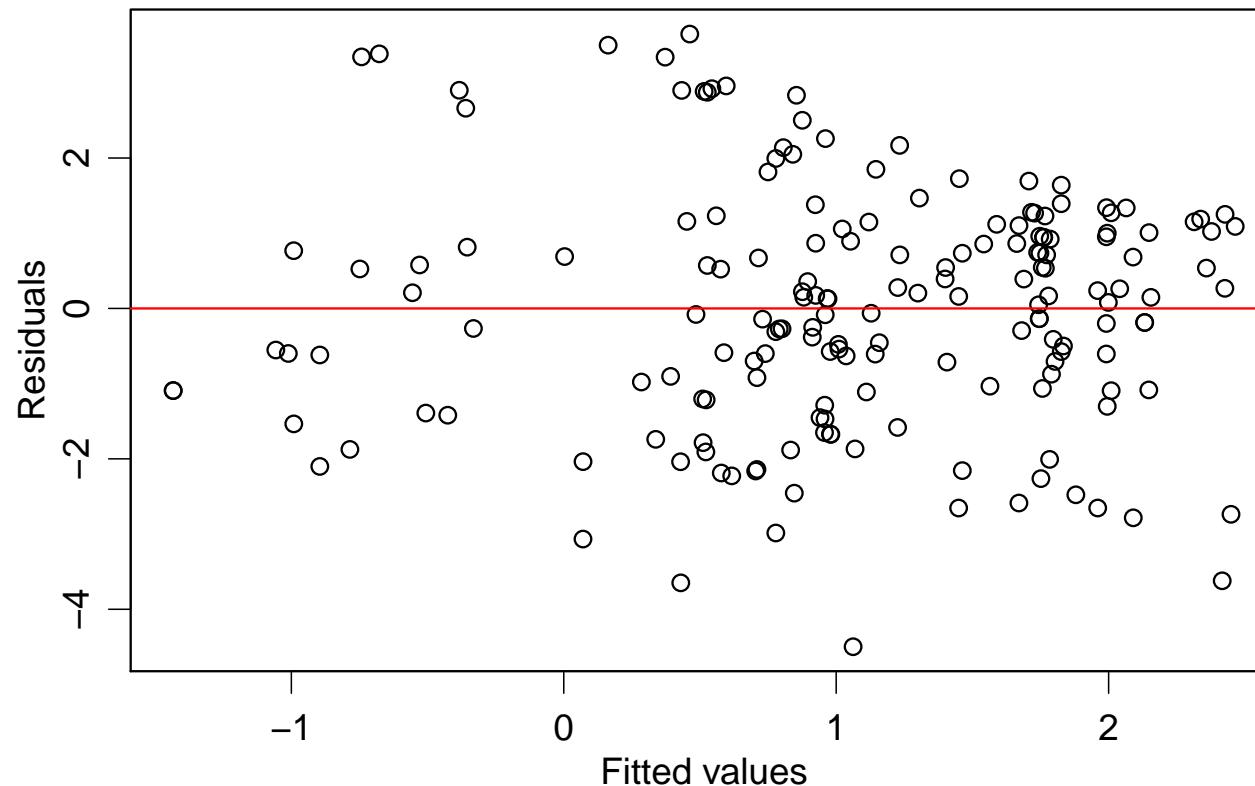
```
> plot(ft.heightgam, residuals=T)
```



The `residuals=T` argument adds “partial residuals” to the smooth plot. It’s not really what we want.

Instead try constructing a residual plot manually:

```
> plot(residuals(ft.heightgam)~fitted(ft.heightgam),  
       xlab="Fitted values",ylab="Residuals")  
> abline(h=0,col="red")
```



What do you reckon?

Did I need to use a smoother?

Was a smoother worthwhile or could I have got away with a linear fit? We want to compare the `gam` to a linear model, and one way to do this (as can be done for any nested models) is to use the `anova` function:

```
> ft.heightlm = lm(log(height)~lat,dat=datheight)
> anova(ft.heightlm,ft.heightgam,test="F")
```

Analysis of Variance Table

Model 1: `log(height) ~ lat`

Model 2: `log(height) ~ s(lat)`

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	176	445.4				
2	176	445.4	2.5665e-10	4.1973e-10	0.6462	2.905e-09 ***

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1						

Was the spline term worth the trouble? Note there's a problem here...look at the Df!!! The `gam()` function (by default) fits a linear model if the relationship is very linear.

Did I need a smoother – model selection

Alternatively, the question of whether to use a smoother could be viewed as a **model selection** problem.

e.g. compare BIC for these two models:

```
> BIC(ft.heightlm,ft.heightgam)
      df      BIC
ft.heightlm  3 683.9463
ft.heightgam 3 683.9463
```

Was the spline term worth the trouble? Also note the df!!!

Or validation on test data:

```
> n=dim(datheight)[1]
> nTrain = n^0.75
> isTrain=sample(n,nTrain)
> datTrain=datheight[isTrain,]
> datTest = datheight[-isTrain,]
> ft.gam = gam(log(height)~s(lat),dat=datTrain)
> ft.lm = lm(log(height)~lat,dat=datTrain)
> pr.lm = predict(ft.lm,newdata=datTest)
> pr.gam = predict(ft.gam,newdata=datTest)
> print( c( sum(log(datTest$height)-pr.lm)^2, sum(log(datTest$height)-pr.gam)^2 ) )
[1] 2379 2379
```

Was the spline term worth the trouble?

Smoothers with interactions

Spline smoothers are sometimes called additive models or **generalised additive models**. But they don't need to be additive – you can include interactions too.

A bivariate spline for x_1 and x_2 on the same scale:

$$s(x_1, x_2)$$

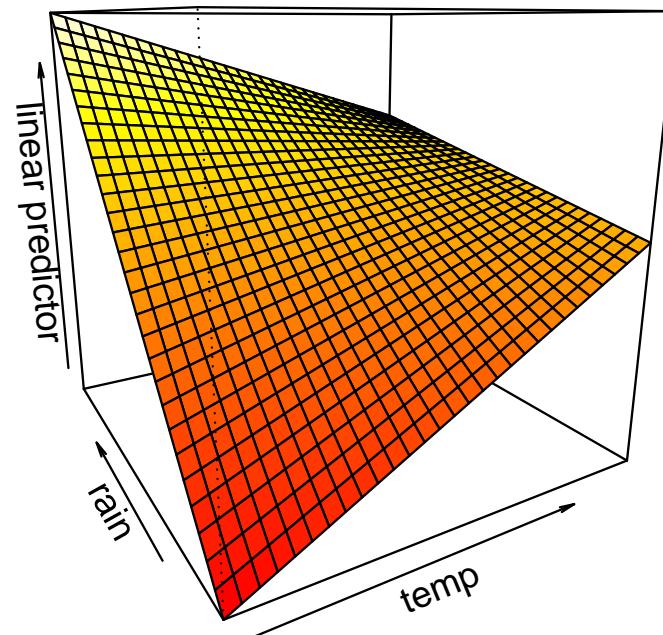
but if not on the same scale:

$$te(x_1, x_2)$$

te for “tensor product smoother”

You can get a 3D plot of what the smoother looks like using the `vis.gam` function.

```
> ft.temprainte = gam(log(height)~te(temp,rain),dat=datheight)
> vis.gam(ft.temprainte,theta=-30)
```



Alternatively, you could try handling interactions in the usual way (quadratic interaction term) with additive smoothers:

```
> ft.temprain = gam(log(height)~s(temp)+s(rain)+rain:temp,dat=datheight)
> summary(ft.temprain)
```

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.11e+00	4.68e-01	4.51	1.2e-05 ***
rain:temp	-3.98e-05	1.71e-05	-2.32	0.021 *

Approximate significance of smooth terms:

	edf	Ref.df	F	p-value
s(temp)	1	1	22.5	4.2e-06 ***
s(rain)	1	1	13.0	4e-04 ***

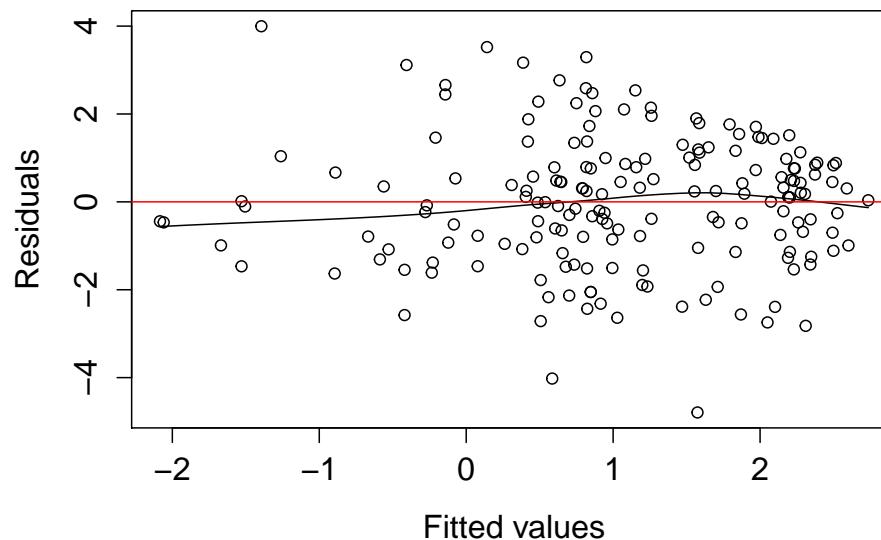
This has the advantage of ensuring only one model coefficient is devoted to the interaction term (# knots quickly gets out of control for interactions).

Any evidence of a rain \times temp interaction?

Smoothers in residual plots

You can plot any two variables, with a smoother, using the `scatter.smooth` function. This is useful as a means to check for a mean trend in manually constructed residual plots:

```
> scatter.smooth(residuals(ft.temprain)~fitted(ft.temprain),  
+                 xlab="Fitted values",ylab="Residuals")  
> abline(h=0,col="red")
```



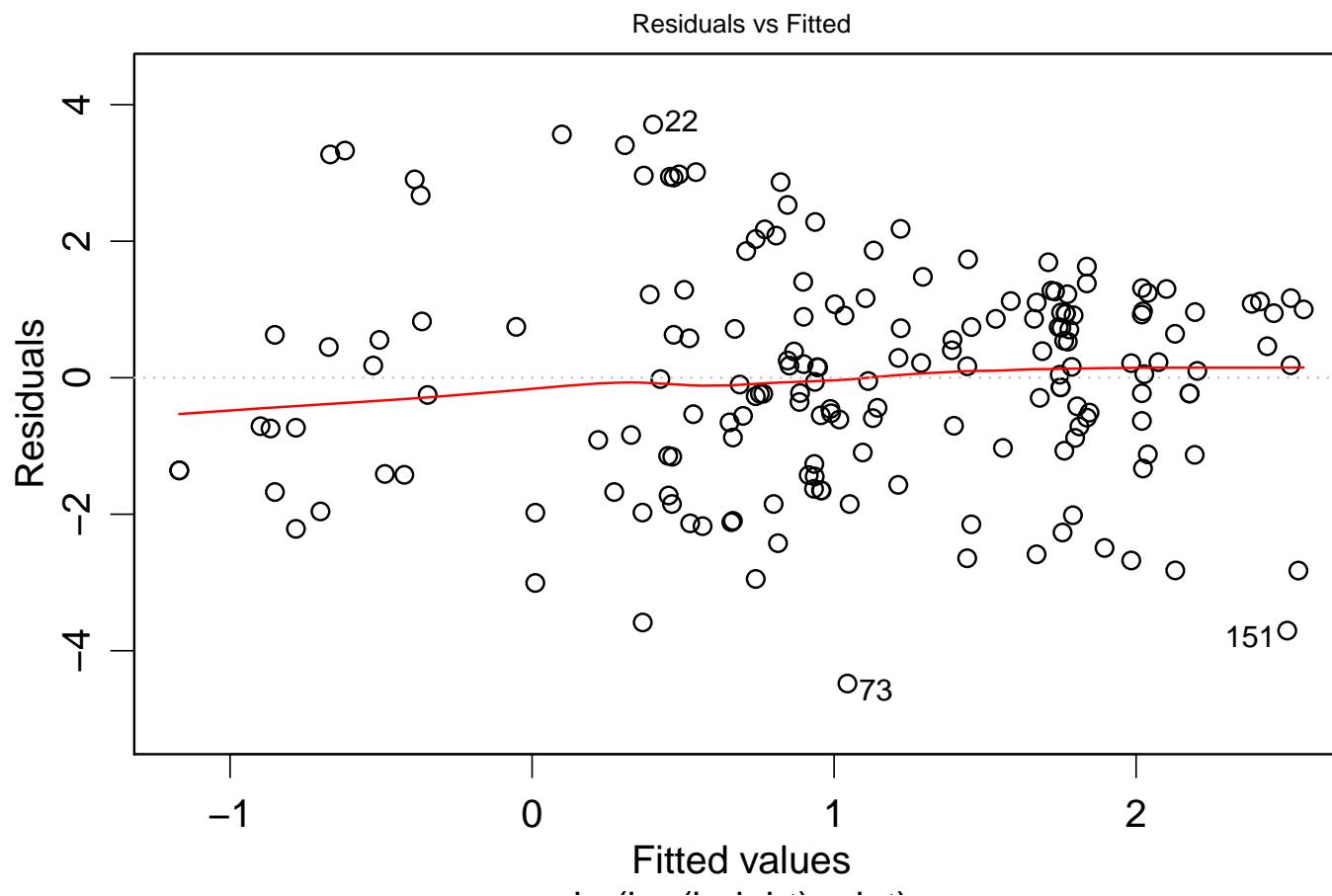
`scatter.smooth` (by default) does not use splines, it uses a different method (local or kernel fitting).

There are actually heaps of different ways to fit smooth curves to data. Splines are the most common in the regression setting because they allow us to stay in the linear modelling framework, with all its benefits (especially diagnostic and inferential tools)

Smoothers on standard residual plots

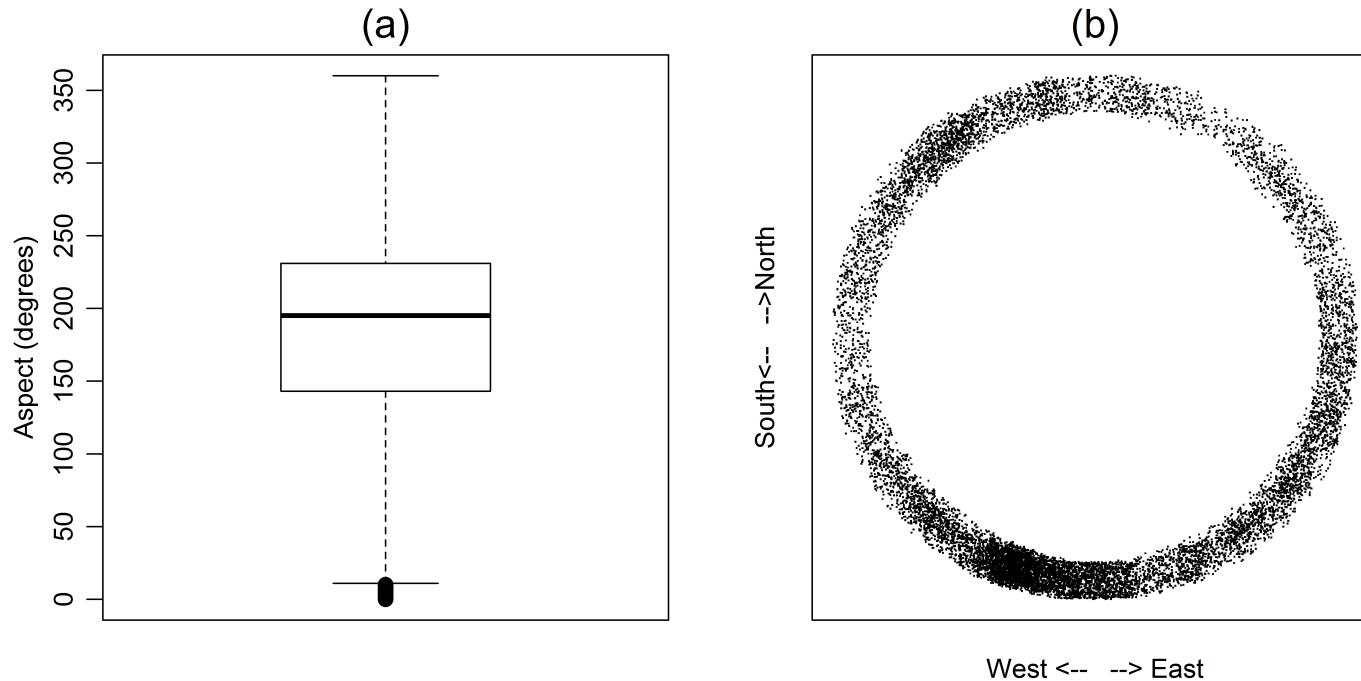
Most residual plot functions on R include a smoother by default. e.g.

```
> ft.heightlm = lm(log(height)~lat,dat=datheight)
> plot(ft.heightlm,which=1)
```



Cyclical variables

Often called “circular variables” or “circular statistics” – variables which are cyclical are more naturally understood by mapping them onto a circle rather a straight line. For example, aspect of slopes on which sheep were found (in degrees, $0 = 360$ =due North):



Which of these graphs makes more sense to you?

How do you map variables onto a circle?

By cos- **and** sin-transforming them (cos was x -axis and sin was y -axis). You have to get the period of the transformation right though – you have to “time” it so that a full cycle has length 2π .

e.g. Transforming aspect (which goes from 0 to 360 degrees):

$$\cos\left(\frac{2\pi \text{ aspect}}{360}\right), \sin\left(\frac{2\pi \text{ aspect}}{360}\right)$$

Transforming time of day (which goes from 0 to 24 hours):

$$\cos\left(\frac{2\pi \text{ time}}{24}\right), \sin\left(\frac{2\pi \text{ time}}{24}\right)$$

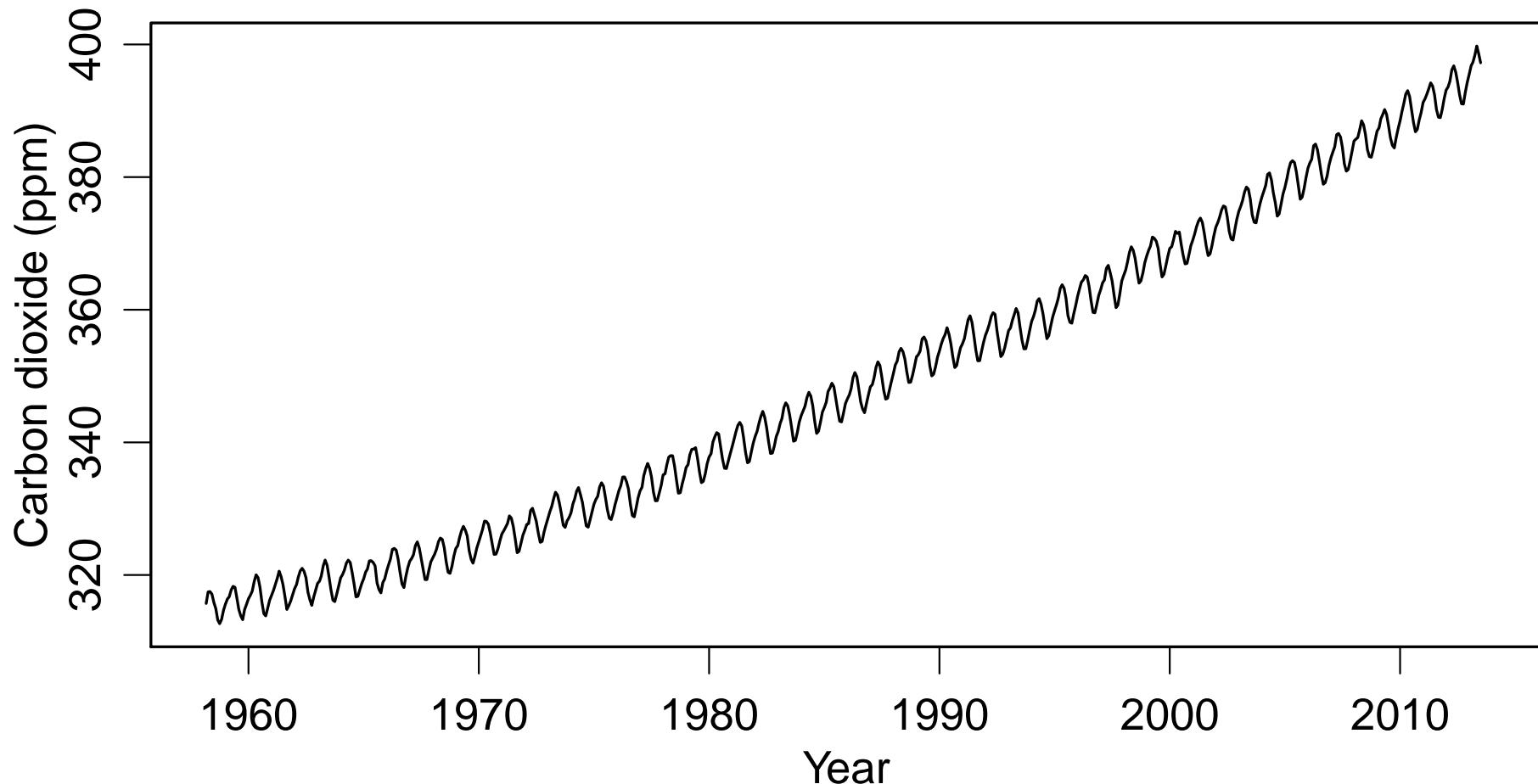
(If time were measured in minutes, you would divide by $24 \times 60 = 1440$.)

Transforming month of year (which goes from 0 to 12 months):

$$\cos\left(\frac{2\pi \text{ month}}{12}\right), \sin\left(\frac{2\pi \text{ month}}{12}\right)$$

Cyclical predictors in linear models

Consider ambient carbon dioxide (ppm) at Mauna Loa observatory:

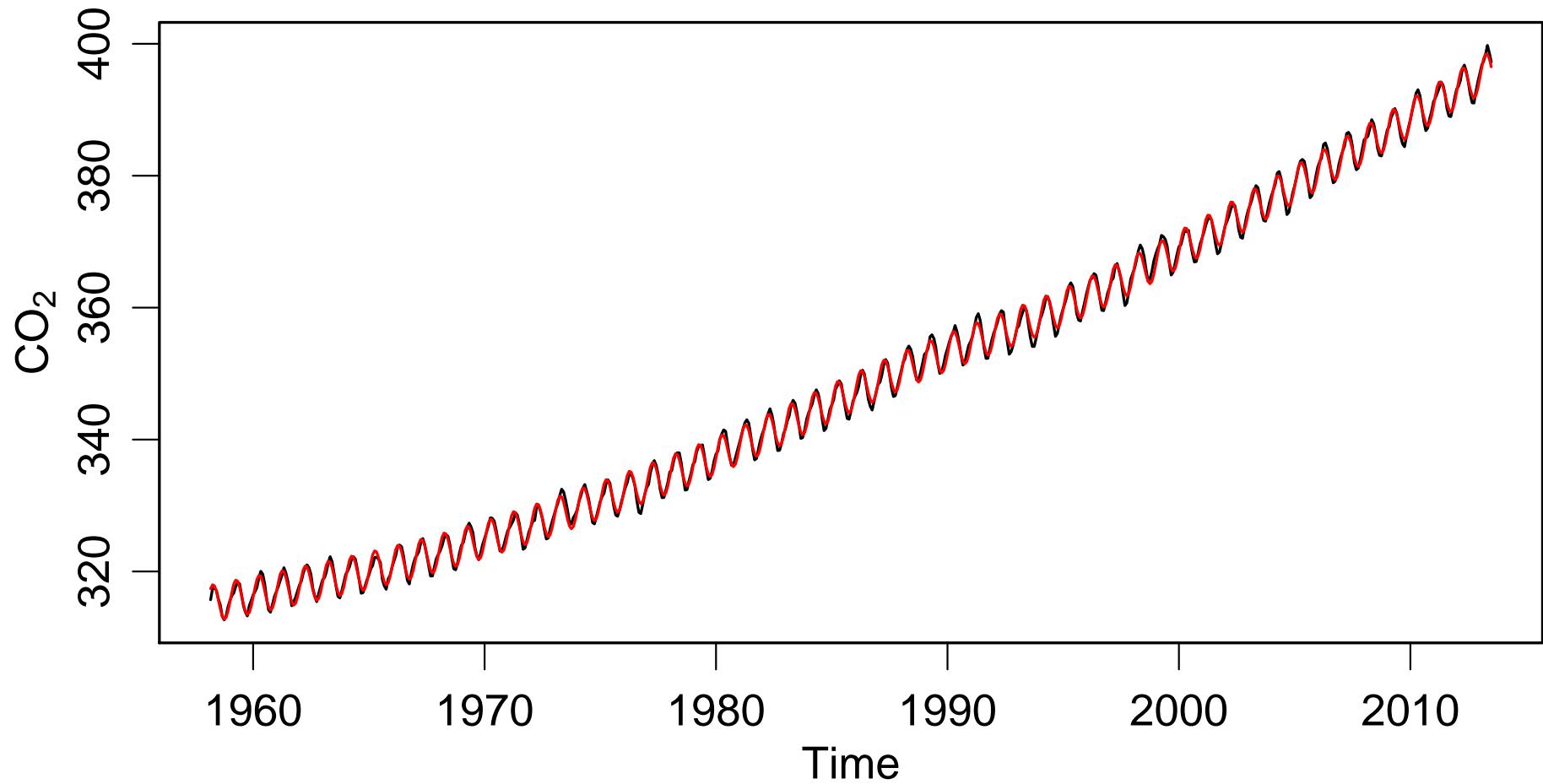


Two things stand out:

- The increasing trend – try `poly` or `gam`
- The periodic wiggles (seasonal variation) – need to include `cos(month)` and `sin(month)`. This adds a **sine curve** to the model.

Cyclical predictors in linear models

```
> ft.cyclic=gam(co2~s(DateNum)+sin(month/12*2*pi)+cos(month/12*2*pi),  
+ data=datmauna)  
> plot(datmauna$co2~datmauna$DateNum,type="l",  
+ ylab=expression(CO[2]),xlab="Time")  
> points(predict(ft.cyclic)~datmauna$DateNum,type="l",col="red")
```

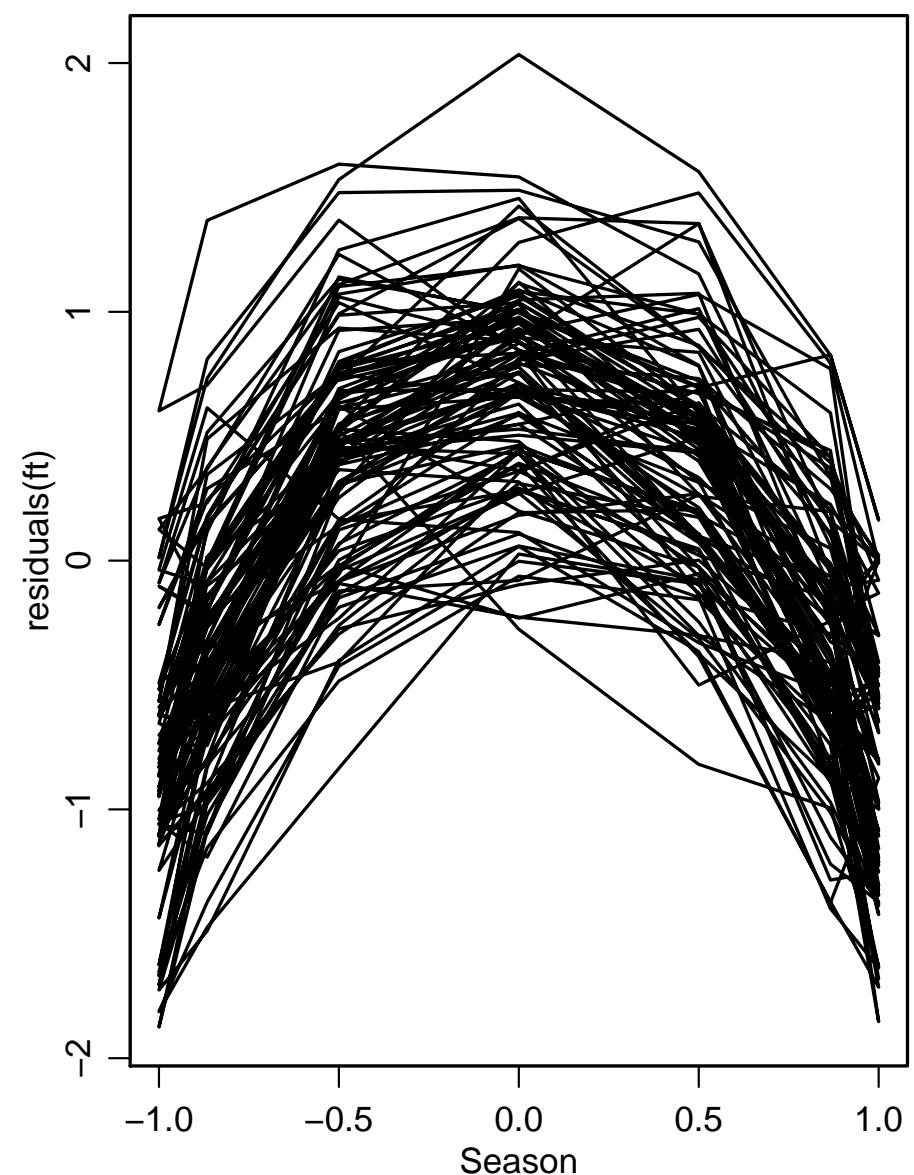
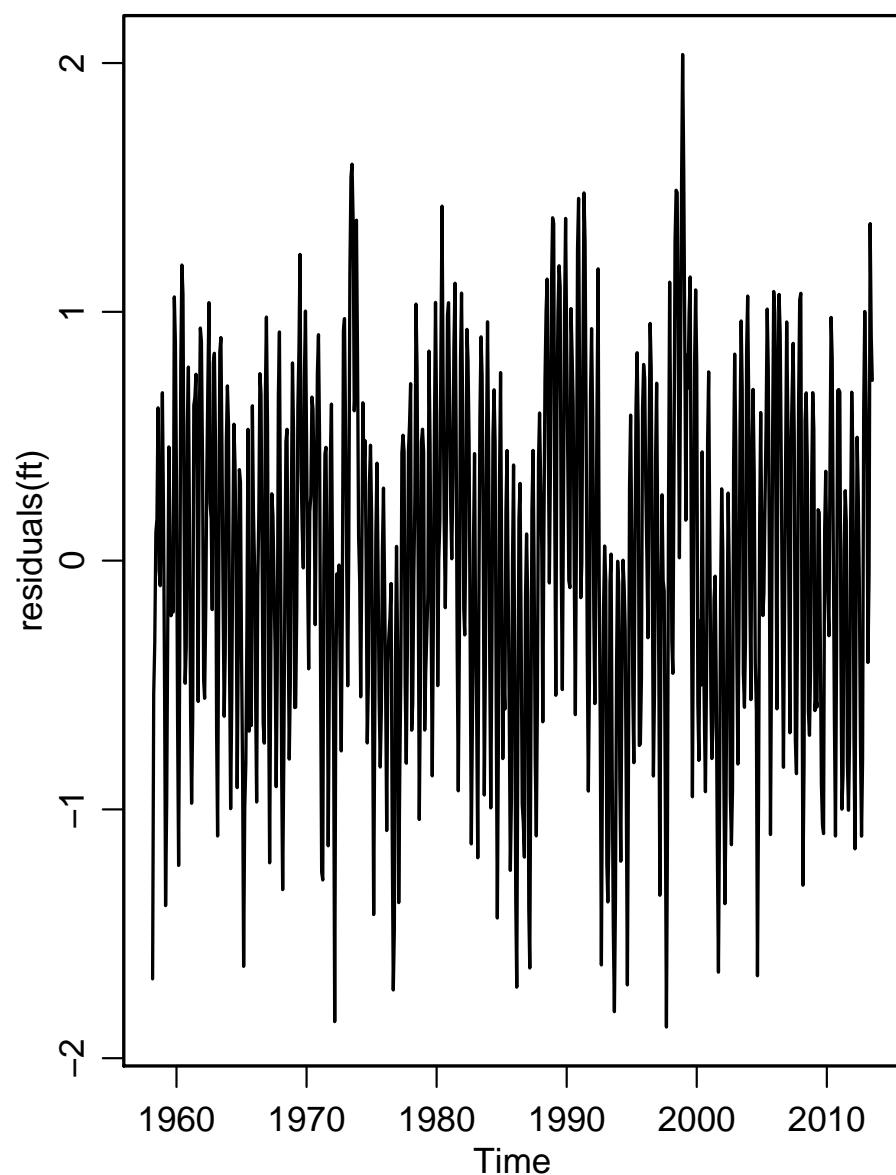


9.29

Diagnosing the model of periodicity

To check we got each component right, we need residual plots against each component. As well as a residual plot against Date (to diagnose smoother) we need a residual plot against season (to diagnose periodicity). Can also do this as a plot against `sin(month/12*2*pi)` (or `cos`).

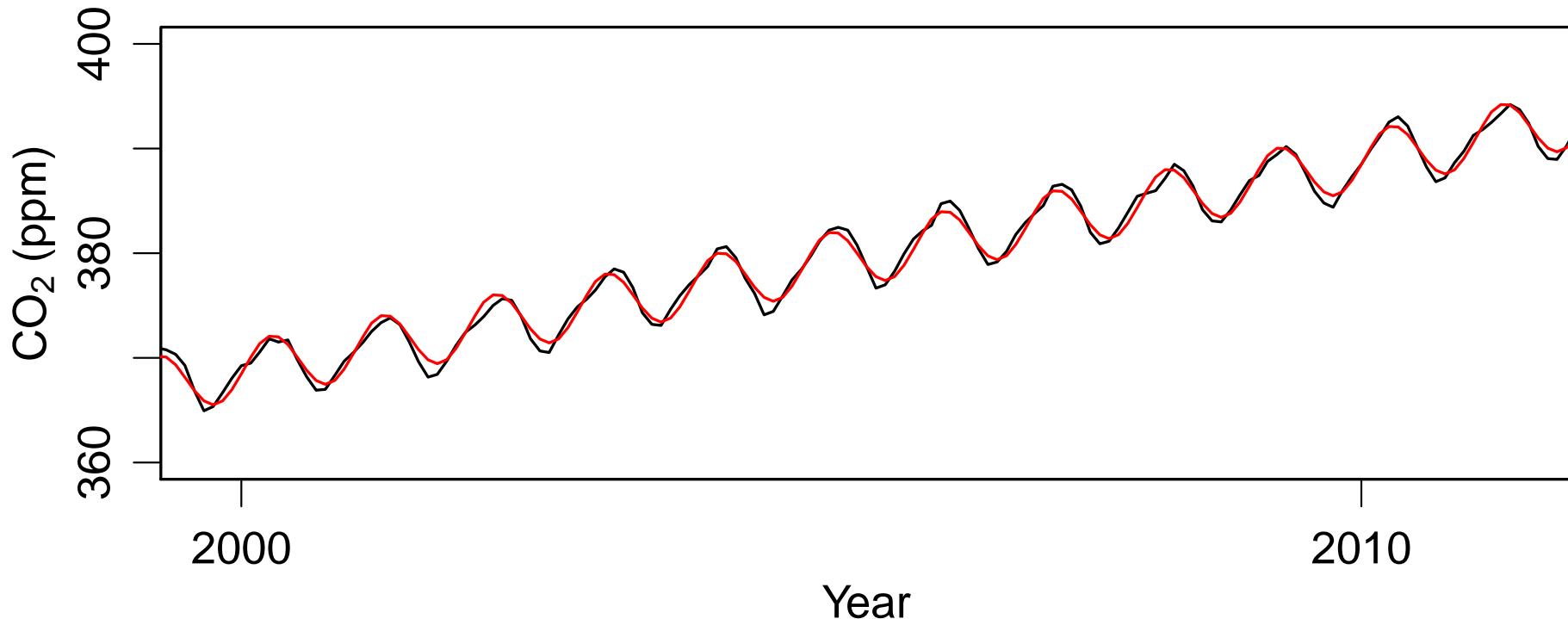
```
> par(mfrow=c(1,2))
> plot(residuals(ft.cyclic)~datmauna$DateNum,type="l",
       xlab="Time")
> plot(residuals(ft.cyclic)~sin(datmauna$month/12*2*pi),
       type="l",xlab="Season")
```



What do you reckon?

Fancier periodic models

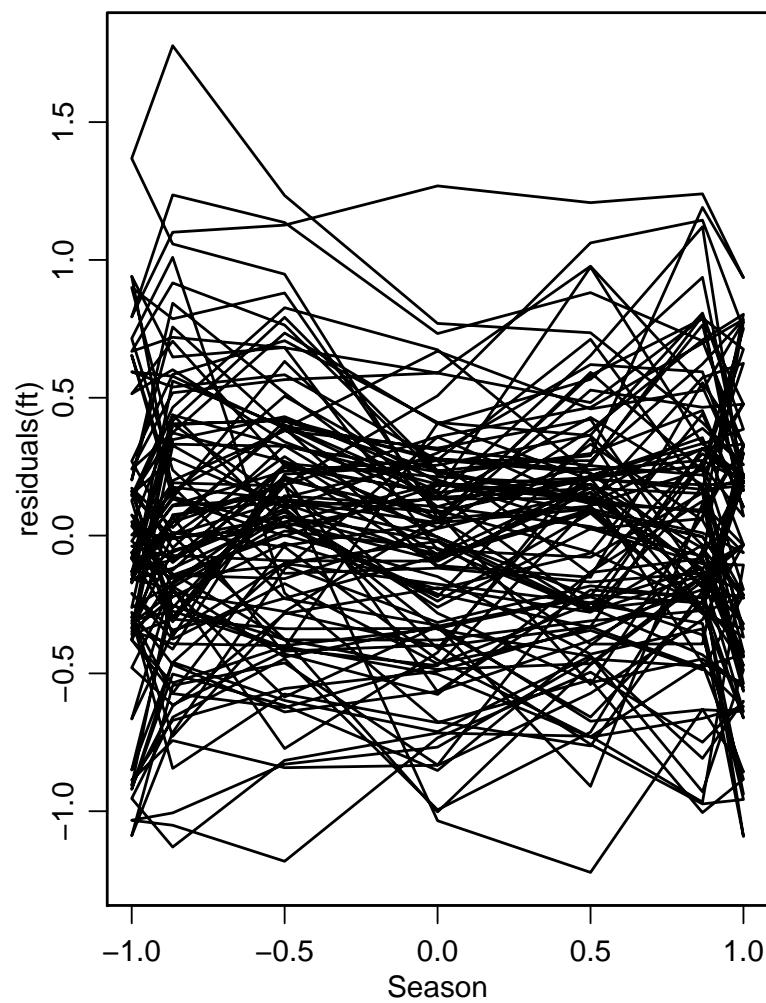
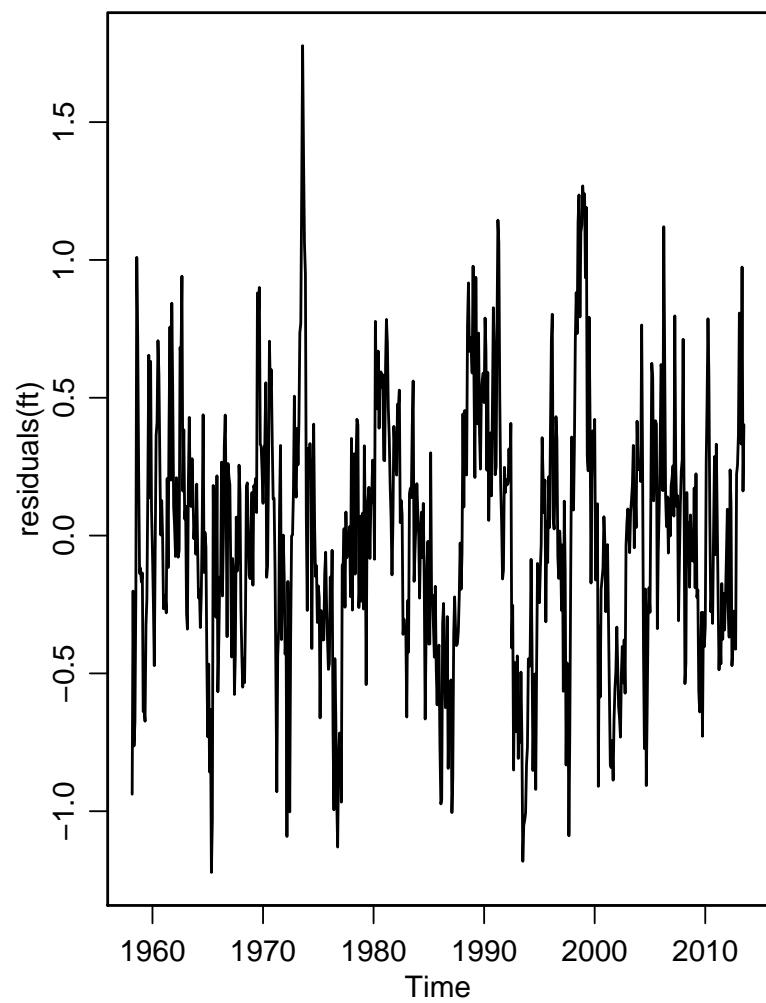
Notice the hump shape on the residual plot against season. This is because a sine-curve did not adequately capture the seasonal trend. You can see this more clearly by zooming in on the 00's:



For quantitative predictors, if a linear trend doesn't work, a simple thing you can try is quadratic terms.

For circular variables, if a sine-curve doesn't work, a simple thing you can try is adding sin- and cos- terms with half the period (the circular world's equivalent of quadratic terms). Or if that doesn't work, a third or a quarter as well!

```
ft.cyclic2=gam(co2~s(DateNum)+sin(month/12*2*pi)+cos(month/12*2*pi)
+sin(month/12*4*pi)+cos(month/12*4*pi),data=datmauna)
```



What do you reckon?

The mean trend seems to have been dealt with, but there is residual correlation between variables (since CO₂ this month depends on CO₂ last month).

This correlation (or “time lag”) becomes a problem for inference – standard errors and *P* values will be too small, differences in BIC too big...

The best option then is to either think about a time series model or random effects with a temporal autocorrelation structure.

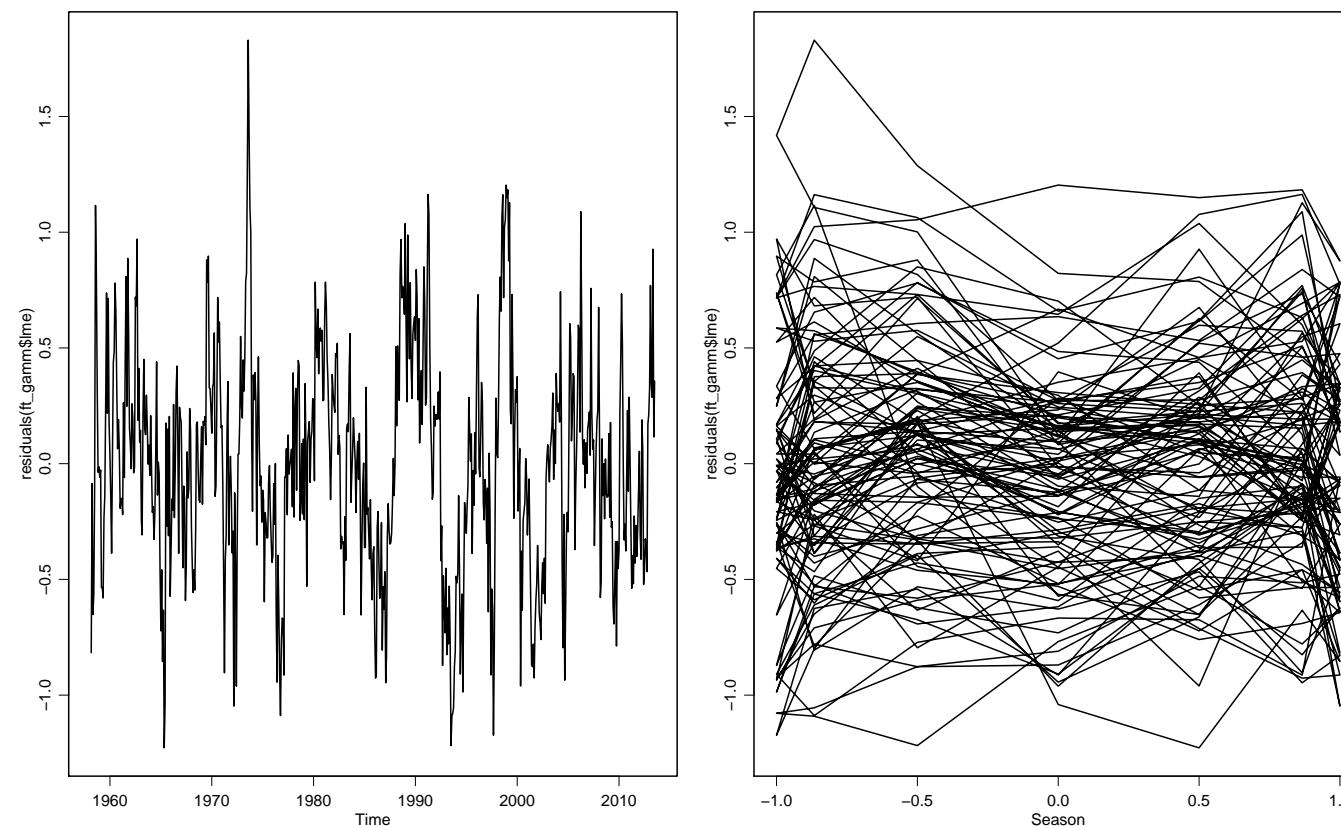
So lets try a random effects model with a temporal autocorrelation structure of order 1.

This structure allows for the random error from the previous year to be correlated and predictive of the random error in the current year. We could also try higher orders.

Two new bits to our function are added:

```
ft.gamm<-gamm(co2~s(DateNum)+sin(month/12*2*pi) + cos(month/12*2*pi)
+ sin(month/12*4*pi) + cos(month/12*4*pi),
correlation=corAR1(form=~1|year),data=datmauna);
```

There isn't much of a difference in the residual plots, but the standard errors are now different for the coefficient estimates!



```
> summary(ft.cyclic2)
```

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	349.01445	0.01845	18919.169	<2e-16	***
sin(month/12 * 2 * pi)	2.25807	0.02610	86.507	<2e-16	***
cos(month/12 * 2 * pi)	-1.66910	0.02609	-63.976	<2e-16	***
sin(month/12 * 4 * pi)	-0.03096	0.02610	-1.186	0.236	
cos(month/12 * 4 * pi)	0.76976	0.02608	29.513	<2e-16	***

```
> summary(ft.gamm$gam)
```

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	349.01413	0.04093	8527.482	<2e-16	***
sin(month/12 * 2 * pi)	2.25264	0.03569	63.125	<2e-16	***
cos(month/12 * 2 * pi)	-1.67095	0.03177	-52.593	<2e-16	***
sin(month/12 * 4 * pi)	-0.03478	0.02037	-1.707	0.0882	.
cos(month/12 * 4 * pi)	0.76928	0.01939	39.677	<2e-16	***

Design-based inference

- Model-based vs design-based inference
- Permutation testing
- Bootstrapping
- Resampling residuals
- What resampling can and can't do

Model-based vs design-based inference

Recall there are three main types of inferential procedures:

- Confidence interval estimation
- Hypothesis testing
- Model selection

In each case we can either take a model-based or a design-based approach to inference.

Model-based inference – assume your model is correct (or nearly correct) and use theory (or sometimes simulation) around your model for inference.

Design-based inference – exploits independent units in your study design as a basis for inference (usually via simulation).

Examples you have already seen

Confidence intervals Our CI formula ($\text{estimate} \pm 2\text{standard error}$) can be model-based or design based, depending how standard errors are estimated. Usually model-based (as in the `confint` function).

Hypothesis testing t tests from `summary` and F -tests from `anova` are model-based. So is the parametric bootstrap.

Model selection AIC is model-based, cross-validation is design-based inference (exploiting independence of training/test observations).

This session is on design-based methods of hypothesis testing.

Model-based vs Design-based: pros and cons

Design-based inference makes less restrictive assumptions – the model does not have to be (nearly) correct in order for inferences to be valid.

Design-based inference almost always takes longer – it is a simulation-based approach. If your model was hard to fit in the first place it will be bloody hard now.

Design-based inference can be less efficient (when assumptions are reasonable) – simulation introduces noise to results, sometimes (especially in model selection) this makes inference less accurate.

Permutation tests

This is applicable whenever the null hypothesis being tested is “no effect of anything” (intercept model).

e.g. Smoking in pregnancy: the number of errors made by guinea pigs in the maze:

C	C	C	C	C	C	C	C	C	N	N	N	N	N	N	N	N	N	N	
11	19	15	47	35	10	26	15	36	20	38	26	33	89	66	23	28	63	43	34

C = Control, N = Nicotine Treatment

If there is no effect of treatment, we can randomly permute Control/Treatment labels without affecting the distribution of the test statistic.

C	C	C	C	C	C	C	C	C	N	N	N	N	N	N	N	N	N	N	
11	19	15	47	35	10	26	15	36	20	38	26	33	89	66	23	28	63	43	34

The observed test statistic is 2.67.

If there is no effect of treatment, we can randomly permute Control/Treatment labels without affecting the distribution of the test statistic.

C	C	C	N	N	N	N	C	C	N	C	C	N	N	C	C	C	N	N	N
11	19	15	47	35	10	26	15	36	20	38	26	33	89	66	23	28	63	43	34

The test statistic for this permutation is 0.76.

If there is no effect of treatment, we can randomly permute Control/Treatment labels without affecting the distribution of the test statistic.

C	N	N	C	C	C	C	C	N	N	C	N	N	N	N	N	N	C	C	
11	19	15	47	35	10	26	15	36	20	38	26	33	89	66	23	28	63	43	34

The test statistic for this permutation is -2.09.

If there is no effect of treatment, we can randomly permute Control/Treatment labels without affecting the distribution of the test statistic.

N	N	N	C	N	N	C	C	C	C	N	N	C	N	C	C	N	C	N	
11	19	15	47	35	10	26	15	36	20	38	26	33	89	66	23	28	63	43	34

The test statistic for this permutation is -1.08.

If you repeat this procedure lots of times (e.g. 1000) you can build up a null distribution for the test statistic (t) – the distribution we would expect t to have across different randomisations of guinea pigs to treatment groups, if there was no effect of treatment.

We can then use this distribution directly to get a P -value – the proportion of permuted samples (or “resamples”) which beat the observed test statistic (*i.e.* how often we expect to beat it by chance).

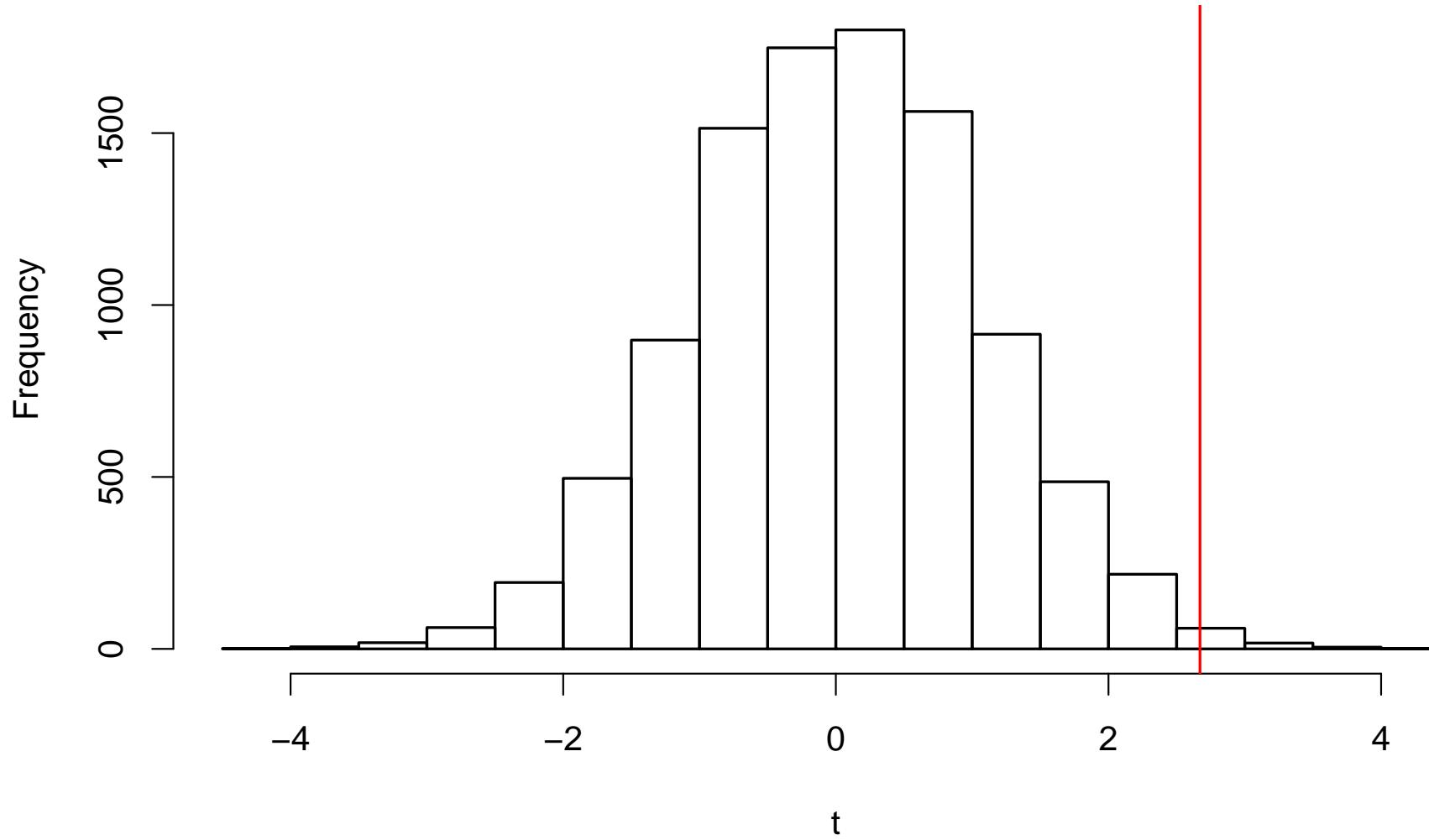
```

> datsmoke=read.csv("smokePregnant.csv")
> datsmokeAll=melt(datsmoke)
Using  as id variables
> colnames(datsmokeAll)=c("treatment","errors")
> ft.smoke = lm(errors~treatment,data=datsmokeAll)
> tObs = summary(ft.smoke)$coef[2,3] #store observed t-statistic
>
> nPerm = 1000
> tStats = rep(NA,nPerm)
> tStats[1] = tObs
> for(iPerm in 2:nPerm)
+ {
+   datsmokeAll$treatPerm = sample(datsmokeAll$treatment) #permute treatment labels
+   ft.smokePerm = lm(errors~treatPerm,data=datsmokeAll) #re-fit model
+   tStats[iPerm] = summary(ft.smokePerm)$coef[2,3] #store t-stat
+ }
> hist(tStats,main="Resampled null distribution of t",xlab="t")
> abline(v=tObs,col="red") #put a red line on the plot for observed t-stat
> p = mean( tStats >= abs(tObs) ) #compute P-value
> print(p)
[1] 0.007

```

Any evidence of an effect of treatment?

Null distribution of t under permutation



That code looks scary...

Well how about this code? (three lines)

```
> library(mvabund)
> ft.smoke = manylm(errors~treatment, data=datsmokeAll)
> anova(ft.smoke, resamp="perm.resid")
Analysis of Variance Table
```

Model: manylm(formula = errors ~ treatment, data = datsmokeAll)

Overall test for all response variables

Test statistics:

	Res.Df	Df.diff	val(F)	Pr(>F)
(Intercept)	19			
treatment	18	1	7.134	0.015 *

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Arguments: with 1000 resampling iterations using residual (without replacement) resampli

The `mvabund` package for resampling-based inference

The `mvabund` package is something we've been developing at UNSW.

`mvabund` stands for “multivariate abundance” data – the main type of data it was designed for (coming this arvo). But it can be used for plenty of other stuff too.

To use `mvabund` for a permutation test of all terms in a linear model:

- Change the function from `lm` to `manylm`
(this helps R work out what to do when you call generic functions like `anova`).
- Add the argument `resamp="perm.resid"` to your `anova` call
(to do permutation testing rather than bootstrapping)

Why are the *P*-values different?

P-value from permutation code – 0.004. From `mvabund` – 0.015.

Two reasons for the differences:

- We are using a **random** set of permutations of treatment labels, so results are random too. The amount of random error (“Monte Carlo error”) in the *P*-value depends on the number of permutations used (controlled by `nBoot` in the `mvabund` package). The more permutations, the smaller the error, `nBoot=1000` (default) is usually good enough.
- The `mvabund` package uses a two-sided test (is there evidence of a **difference** with treatment, not just an **increase**), so it usually gives about double the one-sided *P*-value.

Test stats are also different - `manylm` by default uses an (ANOVA) *F* statistic, which is the square of the *t* stat. ($7.13 \simeq 2.67^2$)

Permutation tests work for regression too

```
> datheight = read.csv("plantHeightSingleSpp.csv")
> ft.height=many lm(height~lat, data=datheight)
> anova(ft.height, resamp="perm.resid")
```

Analysis of Variance Table

Model: many lm(formula = height ~ lat, data = datheight)

Overall test for all response variables

Test statistics:

	Res.Df	Df.diff	val(F)	Pr(>F)
(Intercept)	177			
lat	176	1	13.23	0.001 ***

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Arguments: with 1000 resampling iterations using residual (without replac

How does it compare to model-based P -values?

For linear models, model-based P -values (standard in `lm`) output are **exact** as long as your assumptions are satisfied. *i.e.* the P -value is exactly the probability it claims to be.

If assumptions are violated (especially equal variance) then P -values can be very approximate.

Permutation tests on the other hand are **always exact** (for the hypothesis of “no effect”) even when assumptions aren’t satisfied. Thus some have argued they should be used as a matter of routine.

But they can take longer to compute (especially for large datasets).

Bootstrapping

Another method of design-based inference is the bootstrap. The motivation for the bootstrap is slightly different, but the end result is very similar.

Bootstrapping can be used not just for hypothesis testing, but also for estimating standard errors and confidence intervals.

Bootstrap idea

If we knew the true distribution of our data, we could compute a P -value (/standard error/confidence interval) by simulating values directly from the true distribution. But all we have is our observed data.

The idea of the bootstrap is to use our observed data to estimate the true distribution.

We then **resample** some data – generating a new sample from our best estimate of the true distribution.

e.g. the guinea pig data:

C	C	C	C	C	C	C	C	N	N	N	N	N	N	N	N	N	N	N	N
11	19	15	47	35	10	26	15	36	20	38	26	33	89	66	23	28	63	43	34

If there is no effect of treatment, the 20 observations come from the same distribution.

Without any further assumptions, our best estimate of the true distribution from the observed data is to say it takes the values $11, 19, 15, \dots, 34$ with equal probability ($\frac{1}{20}$).

Example bootstrap samples

Treatment	C	C	C	C	C	C	C	C	C	N	N	N	N	N	N
Obs. data	11	19	15	47	35	10	26	15	36	20	38	26	33	89	66
Bootstrap 1	38	63	26	43	26	43	19	43	89	38	28	38	15	15	33
Bootstrap 2	43	26	26	26	26	23	34	43	63	11	19	35	34	89	43
Bootstrap 3	19	66	28	20	35	38	33	36	26	33	15	43	33	47	23
Bootstrap 4	15	66	28	89	47	10	28	11	19	11	66	89	36	36	47

Bootstrapped data vs permuted data

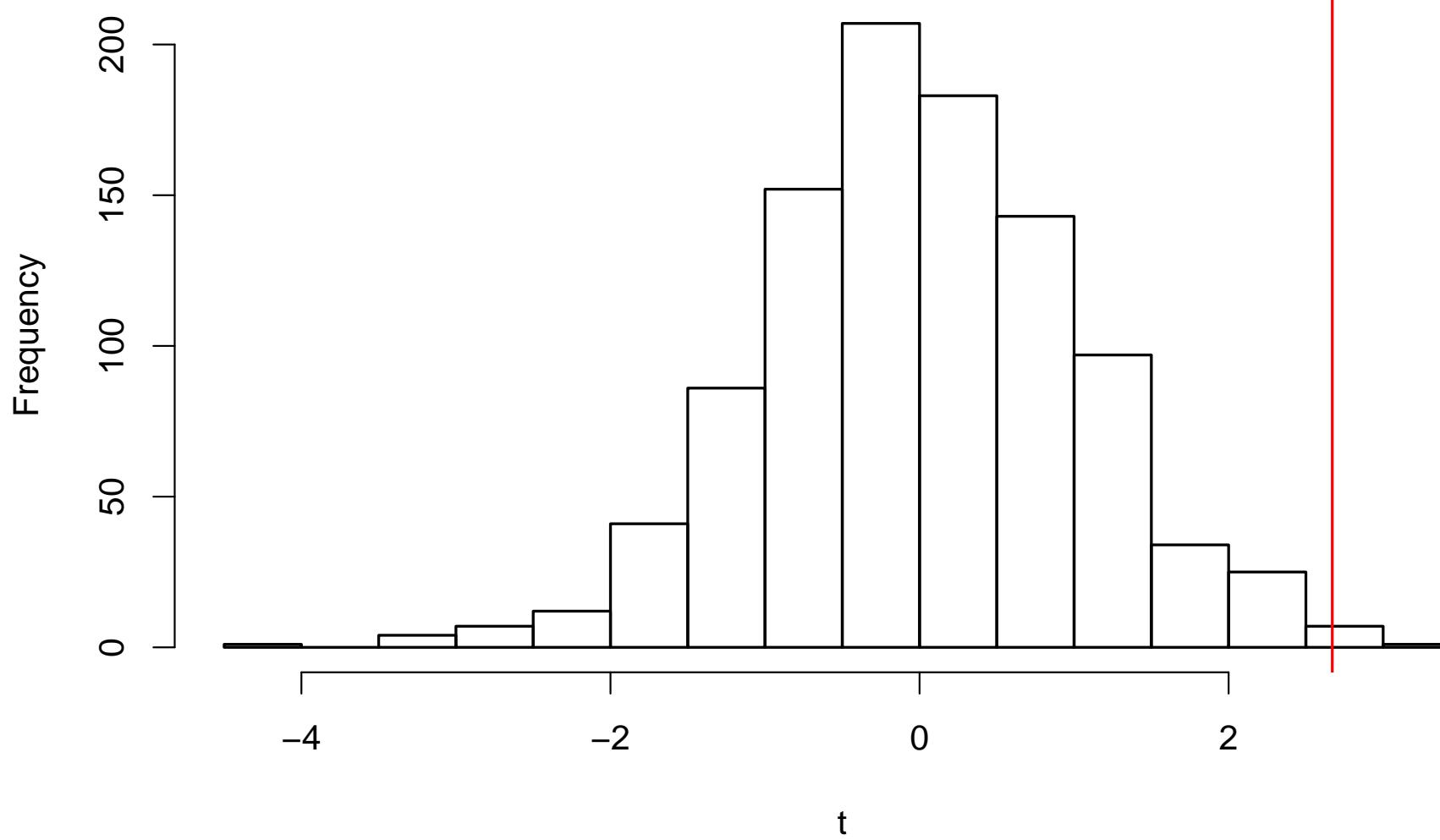
Notice two main differences in the data generation mechanisms for bootstrapping and permutation testing:

- A bootstrap considers the response variable y as random and resamples it, whereas a permutation test considers the treatment labels as random and permutes them. This difference is conceptual but has no practical implication.
- A bootstrap resamples **with replacement** whereas a permutation test uses each response variable exactly once. Again this has little practical implication – results work out basically the same.

Bootstrapped t -test – the ugly version

```
ft.smoke = lm(errors~treatment,data=datsmokeAll)
t0bs = summary(ft.smoke)$coef[2,3] #store observed t-statistic
nBoot = 1000
tStats = rep(NA,nBoot)
tStats[1] = t0bs
for(iBoot in 2:nBoot)
{
  datsmokeAll$errorBoot = sample(datsmokeAll$errors,replace=T) #bootstrap response
  ftBoot = lm(errorBoot~treatment,data=datsmokeAll) #re-fit model
  tStats[iBoot] = summary(ftBoot)$coef[2,3] #store t-stat
}
hist(tStats,main="Resampled null distribution of t",xlab="t")
abline(v=t0bs,col="red") #put a red line on the plot for observed t-stat
p = mean( tStats >= abs(t0bs) ) #compute P-value
print(p)
```

Resampled null distribution of t



10.25

The mvabund package – the pretty version

```
> library(mvabund)
> ft.smoke = manylm(errors~treatment,data=datsmokeAll)
> anova(ft.smoke)
```

Time elapsed: 0 seconds

Analysis of Variance Table

Model: manylm(formula = errors ~ treatment, data = datsmokeAll)

Overall test for all response variables

Test statistics:

	Res.Df	Df.diff	val(F)	Pr(>F)
(Intercept)	19			
treatment	18	1	7.13	0.015 *

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Arguments: with 1000 resampling iterations using residual (without replac

mvabund uses the bootstrap by default

Note we don't need to specify a `resamp` argument – `mvabund` uses the bootstrap by default.

Bootstrap vs permutation test properties

An important practical difference between bootstrapping and permutation testing is that the bootstrap is not wedded to hypothesis testing – you can bootstrap any dataset, whether to test a null hypothesis, to estimate standard errors, do model selection, whatever. The notion of permutation testing does not extend as naturally to these settings.

Whereas permutation tests are exact for “no effect” nulls, no one appears to make such claims for the bootstrap. But if it’s not exact for “no-effect” nulls it must be pretty damn close...

Which one should I use? Either, they work about the same in practice.

Assumptions of resampling methods

Permutation tests assume observations are **exchangeable** (can be swapped around without changing the joint distribution). Technically this is slightly different from assuming independence, but in practice if your observations aren't independent this assumption will almost always be violated.

Bootstrapping assumes observations are **independent**.

There are **no** further assumptions. That's the value of design-based inference – we can make inferences using only independence assumptions which we can guarantee are satisfied via our study design

(how can you guarantee observations are independent?)

Parametric bootstrap

Yesterday we met the parametric bootstrap. The idea there is that we add some assumptions – we assume we know the form of the distribution of our data, and that all we are missing is the true values of parameters. We use our sample estimates of parameters as if they are the true values and then simulate.

Because the parametric bootstrap adds assumptions it works a bit better when the assumptions are satisfied but can be misleading when the assumptions are not satisfied. (This rule about assumptions is a pretty universal one – “no free lunch” principle)

The main point of the parametric bootstrap it is to generate valid P -values when we are reasonably happy with our model, we just don't know how to get good P -values directly from the model e.g. linear mixed models.

Pulling yourself up by the bootstraps

The term “bootstrap” comes from the expression “to pull yourself up by the bootstraps” – suggesting that the bootstrap kind of lets the data get itself out of a sticky situation, without external help (without model-based assumptions). It allows us to make valid inferences from very small samples where previously it was thought you had not a lot to work with.

The bootstrap was a bit of a world-beater in the late '70's early '80's.

Resampling residuals

These methods are all well and good if the goal is to test a hypothesis of no effect – when we can freely resample observations. What if we don't want to do that?

e.g. Angela collects data on plant height and climate characteristics from sites around the world. She wants to know:

Can latitudinal variation in plant height be explained by rainfall?

When the null hypothesis is not “no effect”

The null hypothesis here is that plant height is explained by rainfall. We can't freely permute or bootstrap values – this would break up the data structure under the null (removing the relationships between plant height and rainfall, or between latitude and rainfall).

We want to resample under the null hypothesis that there is a relationship between rainfall and plant height (and possibly between rainfall and latitude).

We do this by **resampling residuals** using the fitted model under the null hypothesis.

$$y_i = \hat{\mu}_i + \hat{\varepsilon}_i$$

1	8.1	-7.1
15	8.1	6.9
7	6.9	0.1
7	6.9	0.1
14	7.4	6.6
4	7.4	-3.4
4	6.1	-2.1
5	6.1	-1.1
2	3.1	-1.1
3	3.1	-0.1
5	1.9	3.1
0	1.9	-1.9
1	1.9	-0.9
1	1.9	-0.9
2	0.6	1.4
1	0.6	0.4

$$= \hat{\mu}_i + \hat{\varepsilon}_i$$

8.1 -7.1

8.1 6.9

6.9 0.1

6.9 0.1

7.4 6.6

7.4 -3.4

6.1 -2.1

6.1 -1.1

3.1 -1.1

3.1 -0.1

1.9 3.1

1.9 -1.9

1.9 -0.9

1.9 -0.9

0.6 1.4

0.6 0.4

← resample
residuals

$$= \hat{\mu}_i + \hat{\varepsilon}_i^*$$

8.1 1.4

8.1 -1.1

6.9 3.1

6.9 -1.1

7.4 -1.1

7.4 -2.1

6.1 -1.1

6.1 6.6

3.1 3.1

3.1 0.1

1.9 3.1

1.9 3.1

1.9 0.1

1.9 6.6

0.6 1.4

0.6 -1.1

← resampled
residuals

$$y_i^* = \hat{\mu}_i + \hat{\varepsilon}_i^*$$

9.5	8.1	1.4
7	8.1	-1.1
10	6.9	3.1
5.8	6.9	-1.1
6.2	7.4	-1.1
5.2	7.4	-2.1
5	6.1	-1.1
12.8	6.1	6.6
6.2	3.1	3.1
3.2	3.1	0.1
5	1.9	3.1
5	1.9	3.1
2	1.9	0.1
8.5	1.9	6.6
2	0.6	1.4
-0.5	0.6	-1.1

← resampled
residuals

Residual resampling is the default in mvabund

```
> ft.heightRL=manyLM(height~rain+lat, data=datheight)
> anova(ft.heightRL, resamp="perm.resid")
Analysis of Variance Table
```

Model: manyLM(formula = height ~ rain + lat, data = datheight)

Overall test for all response variables

Test statistics:

	Res.Df	Df.diff	val(F)	Pr(>F)
(Intercept)	177			
rain	176	1	28.648	0.001 ***
lat	175	1	0.326	0.550

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Arguments: with 1000 resampling iterations using residual (without replac

Can the latitude effect be explained by rainfall?

Residual resampling works for any linear model

```
> dathabconf=read.csv("HabitatConfig.csv")
> dathabconf$Dist = factor(dathabconf$Dist)
> ft.habconf=manylm(log(Total)~Time*Dist*Size, data=dathabconf)
> anova(ft.habconf)
```

Overall test for all response variables

Test statistics:

	Res.Df	Df.diff	val(F)	Pr(>F)
(Intercept)	56			
Time	55	1	0.63	0.443
Dist	53	2	8.31	0.002 **
Size	52	1	2.32	0.131
Time:Dist	50	2	2.52	0.089 .
Time:Size	49	1	0.15	0.696
Dist:Size	47	2	0.35	0.712
Time:Dist:Size	45	2	2.22	0.118

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Arguments: with 1000 resampling iterations using residual (without replacement) resampling and response a

But mvabund doesn't handle random effects, yet...

Assumptions of residual resampling

Residual resampling makes a few extra assumptions as compared to resampling the original observations.

In residual resampling, we assume either:

- that residuals are exchangeable (if permuting them)
- that residuals are independent (if bootstrapping them)

We do not make any assumptions about the shape of the distribution of residuals, but in all other respects, we need our model to be correct for inferences to be valid.

Assumptions of linear models

1. The observed y values are **independent** (after conditioning on x)
2. The y values are **normally distributed** with **constant variance**

$$y \sim \mathcal{N}(\mu_y, \sigma^2)$$

3. **straight line relationship** between mean of y and each x

$$\mu = \beta_0 + \mathbf{x}^T \boldsymbol{\beta}$$

Assumptions of residual resampling of linear models

1. The observed y values are **independent** (after conditioning on x)
2. The y values have **constant variance**

$$y \sim \mathcal{F}(\mu_y, \sigma^2)$$

for some distribution \mathcal{F} with additive errors.

3. **straight line relationship** between mean of y and each x

$$\mu_y = \beta_0 + \mathbf{x}^T \boldsymbol{\beta}$$

Residual resampling relaxes the normality assumption – and that's it!

Basically, the only assumption we have been able to relax is **normality** – which was the least important assumption in the first place!

So residual resampling isn't a solution to all the world's problems.

(It is also a stretch to call it design-based inference – a model is needed to compute residuals!)

What resampling can and can't do

Importance of using the right model/test stat

We have seen that under resampling, the validity of P -values (or se's or CI's or ...) rests primarily on the independence assumption – there is no longer an assumption about the actual distribution of y . Although under residual resampling, we also require linearity and constant variance.

But we still should check linear model assumptions even when they aren't important for the test to be valid.

Why check linear model assumptions that are not needed when resampling?

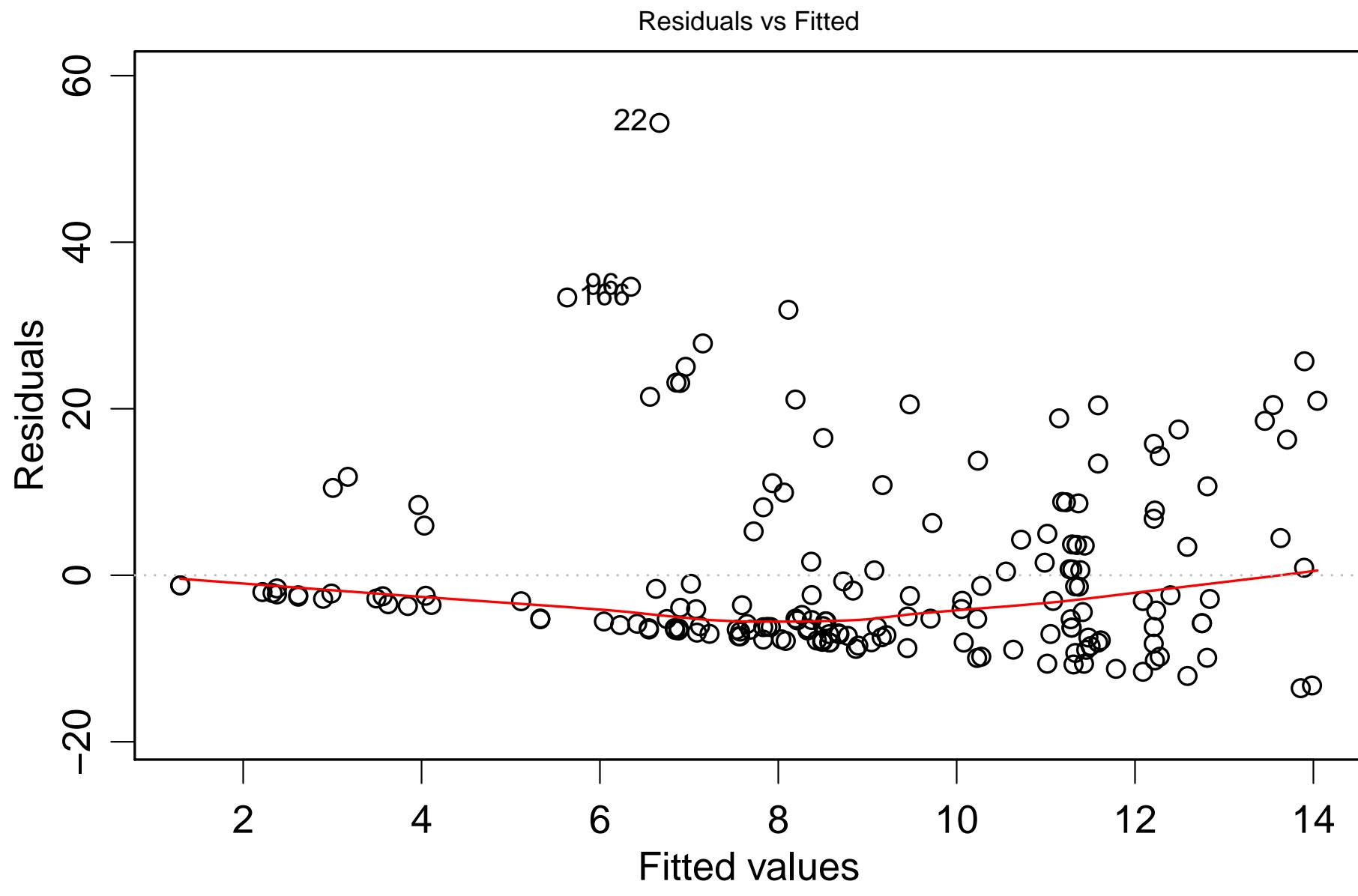
valid \neq efficient

Just because a method is valid doesn't mean it works well (*i.e.* has small se's, high power). Linear models are more efficient (more power) when our assumptions are closer to being satisfied – so try to satisfy them as closely as you can to get the best answer you can!

Example – testing latitude effect

Consider the plant height data, where we were testing for an effect of latitude:

```
> datheight = read.csv("plantHeightSingleSpp.csv")
> ft.height = lm(height~lat, data=datheight)
> plot(ft.height, which=1)
```



What do you reckon?

We probably should have log-transformed our data.

```
> ft.lheight=many lm(log(height)~lat, data=datheight)
> plot(ft.lheight, which=1)
> anova(ft.lheight, resamp="perm.resid")
Analysis of Variance Table
```

Model: many lm(formula = log(height) ~ lat, data = datheight)

Overall test for all response variables

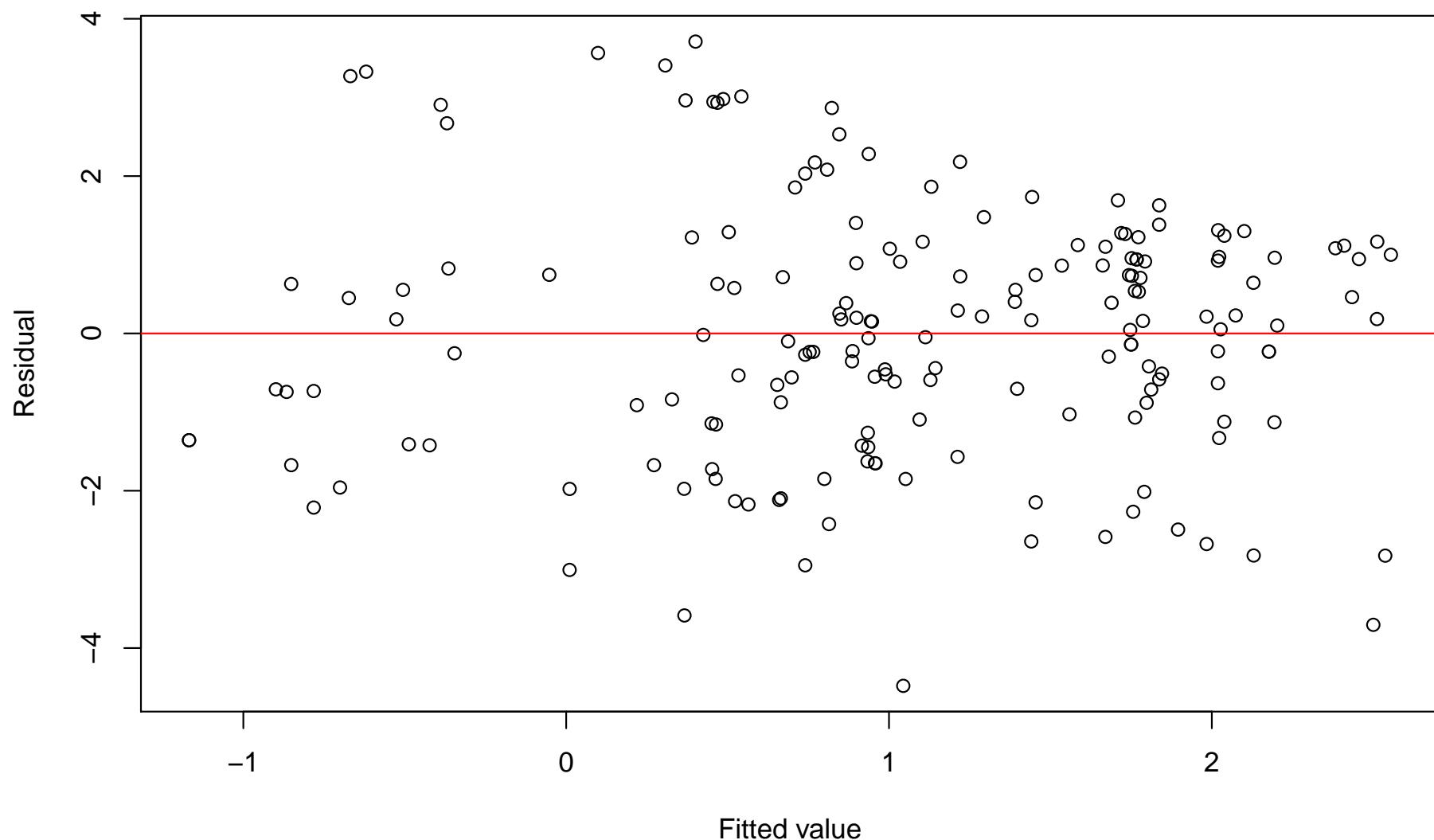
Test statistics:

	Res.Df	Df.diff	val(F)	Pr(>F)
(Intercept)	177			
lat	176	1	53.42	0.001 ***

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Arguments: with 1000 resampling iterations using residual (without replac

Did this affect results?



What do you reckon?

```
> ft.lheightRL=many lm(log(height)~rain+lat, data=datheight)
> anova(ft.lheightRL, resamp="perm.resid")
Analysis of Variance Table
```

Model: many lm(formula = log(height) ~ rain + lat, data = datheight)

Overall test for all response variables

Test statistics:

	Res.Df	Df.diff	val(F)	Pr(>F)	
(Intercept)	177				
rain	176	1	53.20	0.001 ***	
lat	175	1	13.34	0.002 **	

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Arguments: with 1000 resampling iterations using residual (without replac

How do these results compare to last time around?

Take home message

Resampling can make your method valid even when assumptions fail.

But to get a method which works well – has good power, small se's, etc – you need to use a good model (or a good test statistic) for the data at hand. Only with a good model for your data can you ensure that you have a good statistic for answering the research question of interest.

You can get completely different results from different analyses even with resampling (e.g. log-transformed vs untransformed) because of big differences in how reasonable your fitted model is (hence how good your test statistic is).

Exact vs approximate tests

When permuting data under the “no effect” hypothesis, permutation testing is exact.

Under residual resampling, permuting and bootstrapping are **only approximate**. Two tips for ensuring this approximation is good:

- Make sure you resample residuals from a plausible null model. If your null model is wrong (e.g. forgot to transform data) your P -values can be wrong and the whole thing can be invalid. i.e. check assumptions.
- Only estimate the resampling distribution of a standardised (or “pivotal”) statistic, e.g. t -stat, Z -stat, likelihood ratio stat. Do **not** estimate the resampling distribution of an unstandardised statistic (e.g. $\hat{\beta}$). Resampling is known **not to help** improve validity for unstandardised statistics (as compared to using standard distributions – Z , t , etc).

Generalised linear modelling

- Examples
- GLMs – relaxing linear modelling assumptions
- Fitting and checking GLMs
- Inference from generalised linear models
- Extensions

Examples: Crab presence/absence

Alistair is particularly interested in crabs. He observed the following **presence/absence** patterns for crabs (across 10 replicates):

Time	Distance (m)	Crabs
5	0	Absent
5	0	Present
5	0	Present
5	0	Absent
5	0	Absent
5	2	Absent
5	2	Absent
5	2	Present
5	2	Absent
:	:	:
10	10	Present

Is there any evidence of a difference in crab presence patterns with Time or Distance of Isolation?

Examples: Revegetation counts

Anthony wants to evaluate how well invertebrate communities are re-establishing following bush regeneration efforts. Here are some worm counts from pitfall traps across each of 10 sites:

Treatment	C	R	R	R	C	R	R	R	R	R
Count	0	3	1	3	1	2	12	1	18	0

(C=control, R=bush regen)

Is there any evidence that bush regeneration (revegetation) is working?

Discrete data

Something that is different about the above datasets compared to what we saw previously is that the response variable is **discrete**.

continuous: quantitative data that can take any value in some interval
⇒ **linear models**

discrete: quantitative data that takes a “countable” number of values
(e.g. 0, 1, 2, ...) ⇒ **generalised linear models (GLMs)**

If your data are discrete but the counts are all fairly large, you can ignore the discreteness and use linear models anyway. If you have small counts and zeros though it is very important to use GLMs instead.

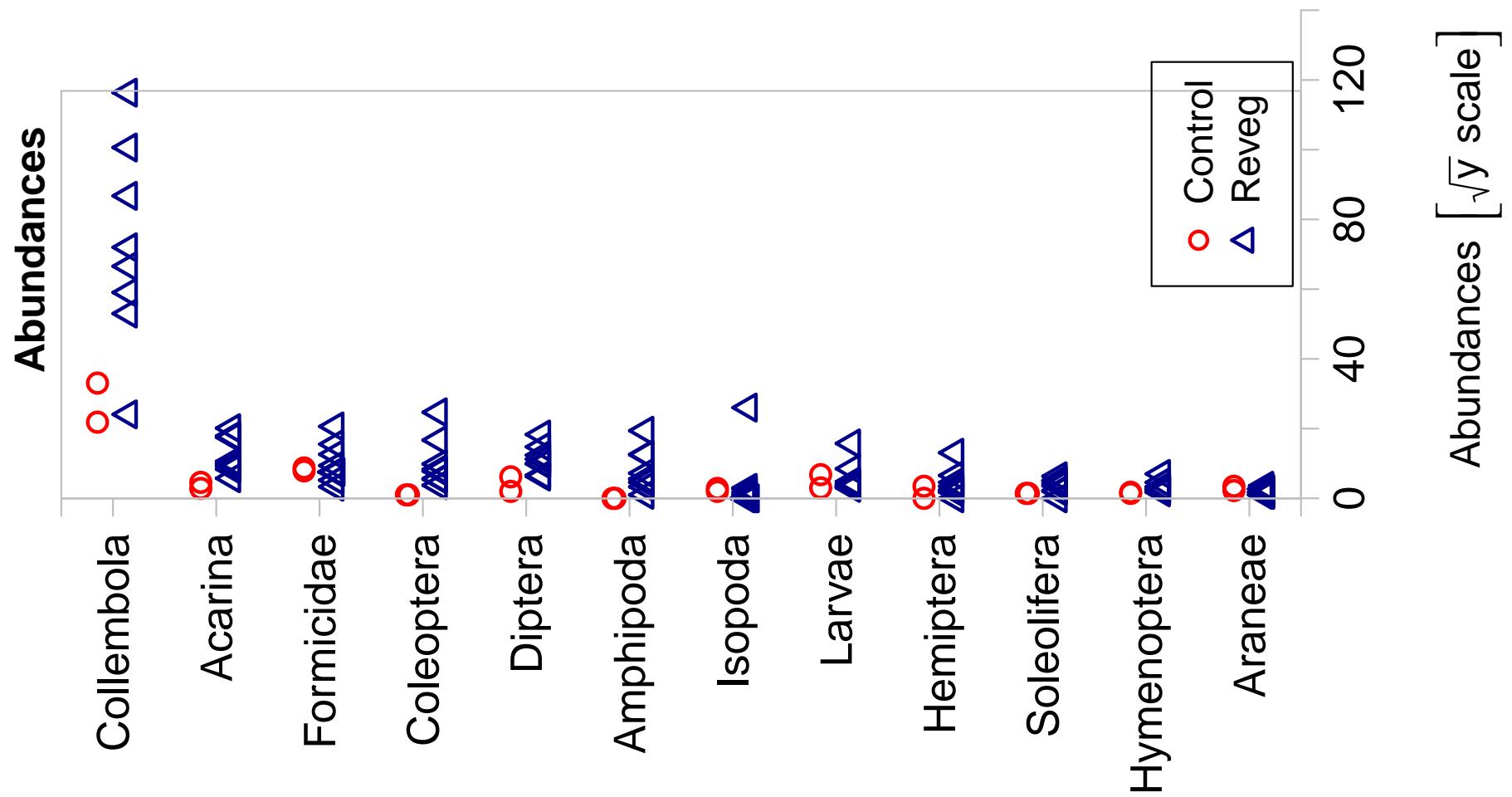
Why does discreteness matter?

The main reason it matters is because it tends to induce a **mean-variance relationship** – as the mean changes the variance changes. When you have zeros and small counts you will have trouble getting rid of the mean-variance relationship via transformation.

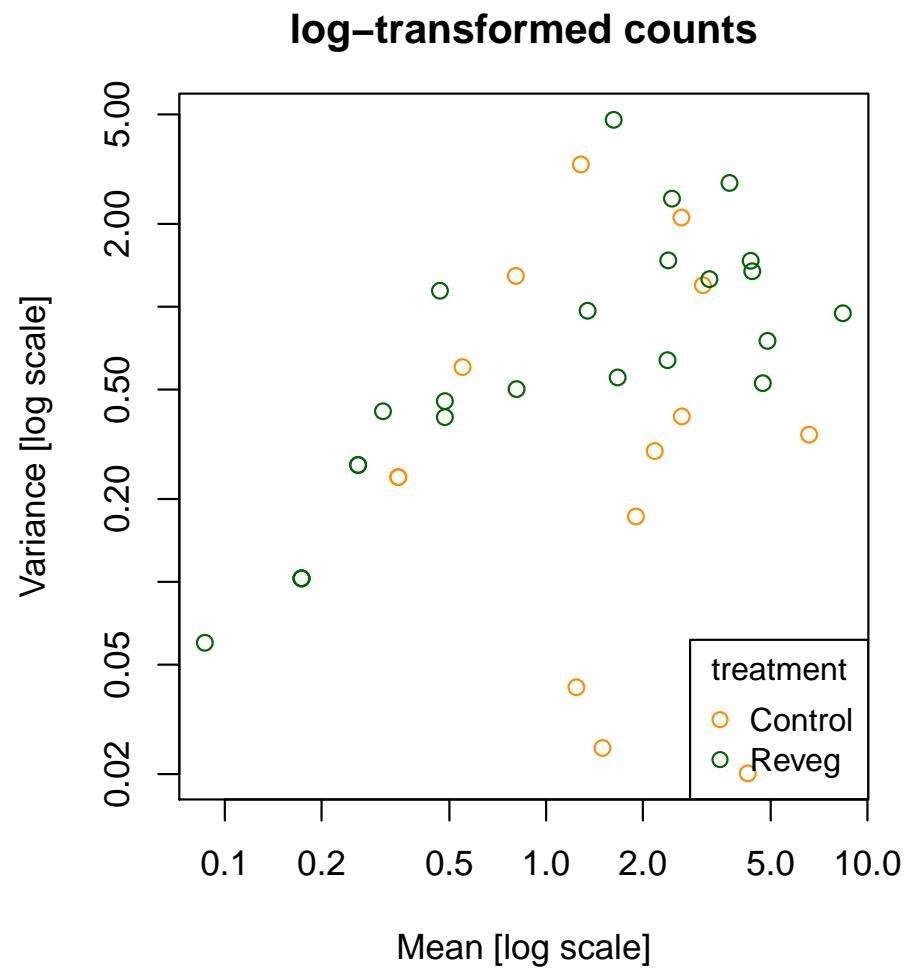
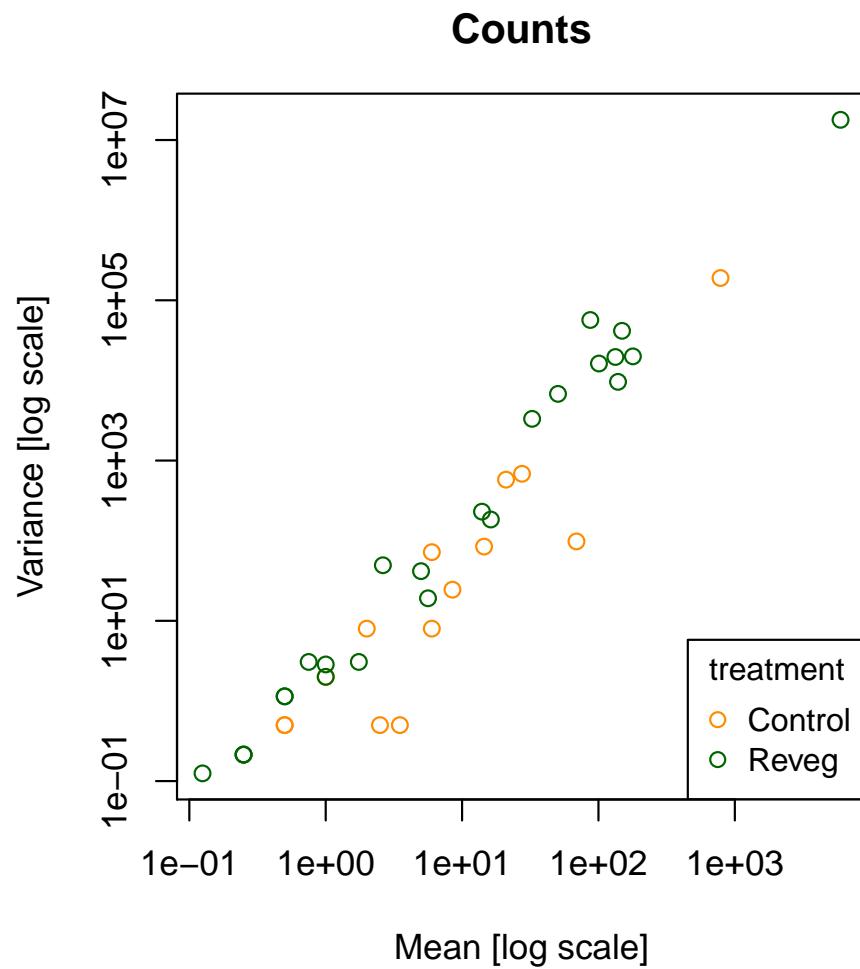
This violates the linear model assumption of constant variance.

Why does it happen? Boundaries. In ecology, it commonly happens because your response variable is hardly ever present in some types of locations (e.g. many species are rare)

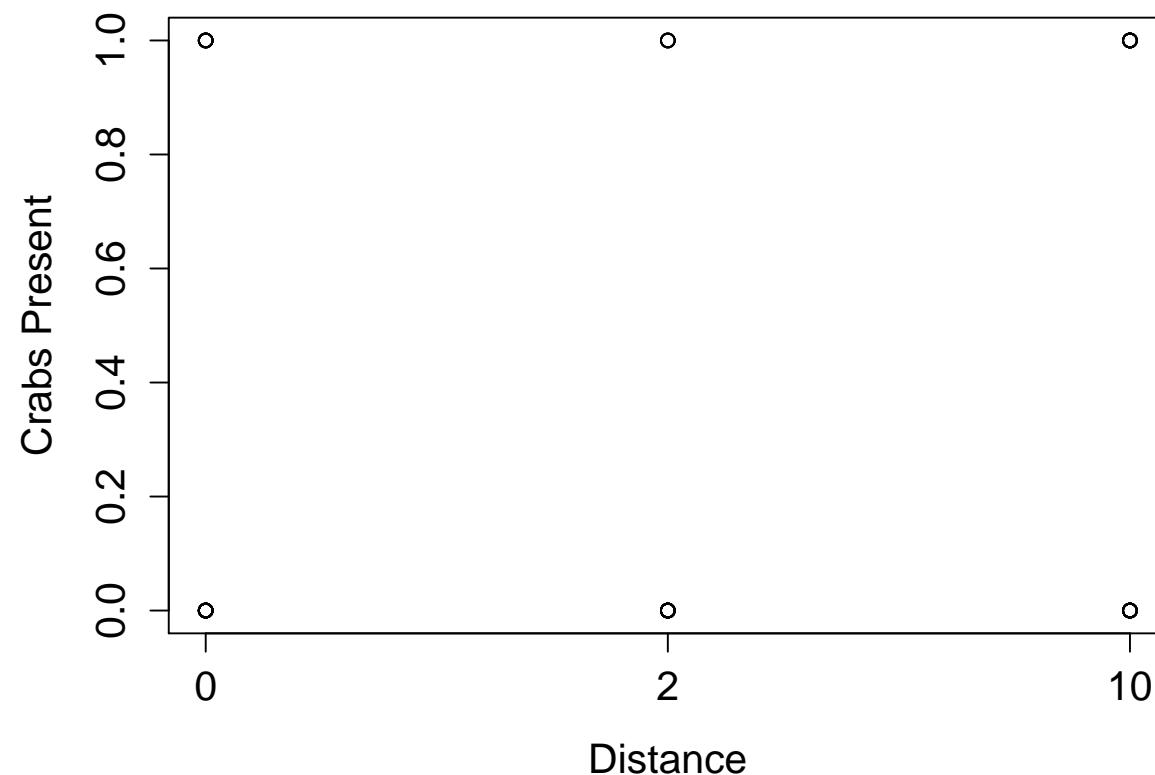
Anthony's revegetation count data (all taxa):



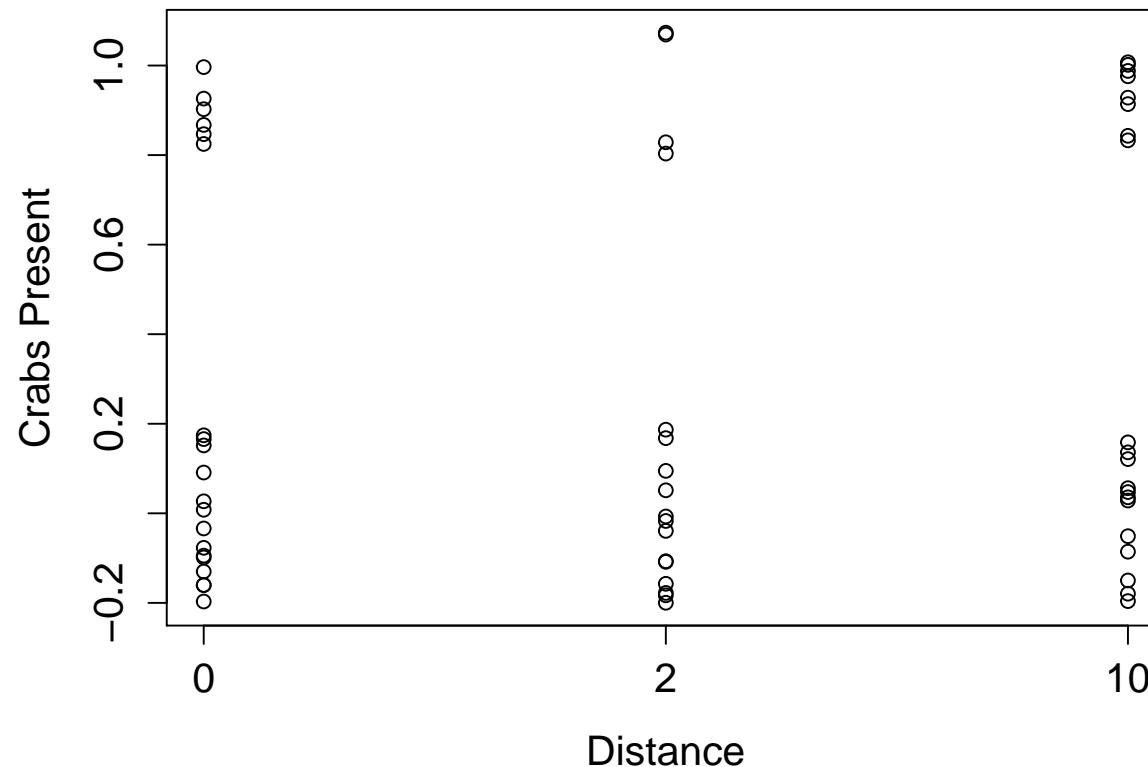
Anthony's mean-variance relationship (transformed or not):



Alistair's crab presence/absence data...overlapping points!

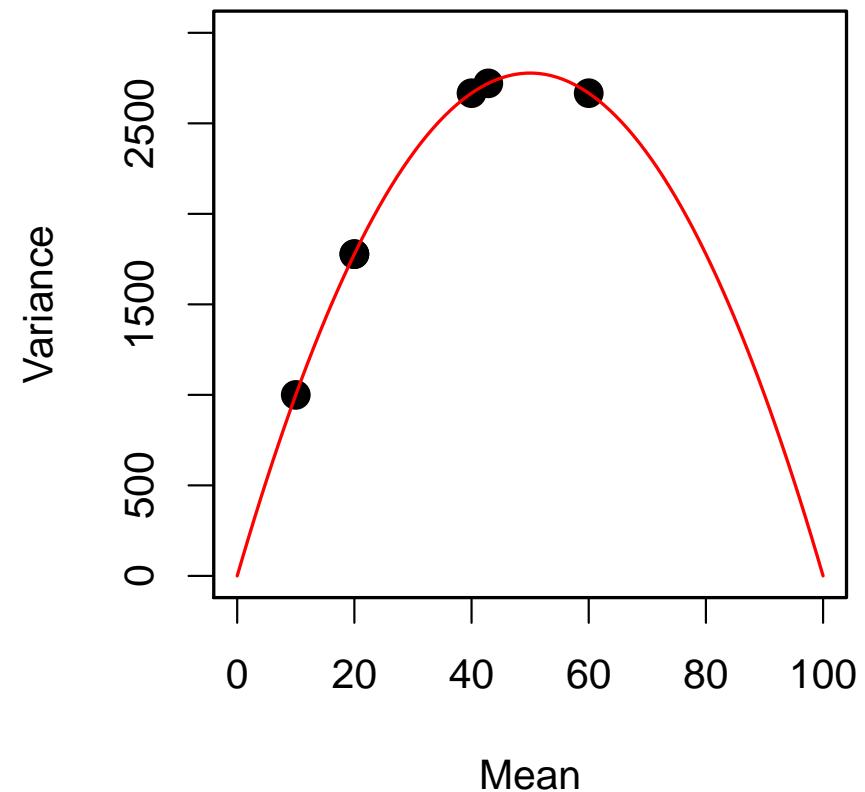
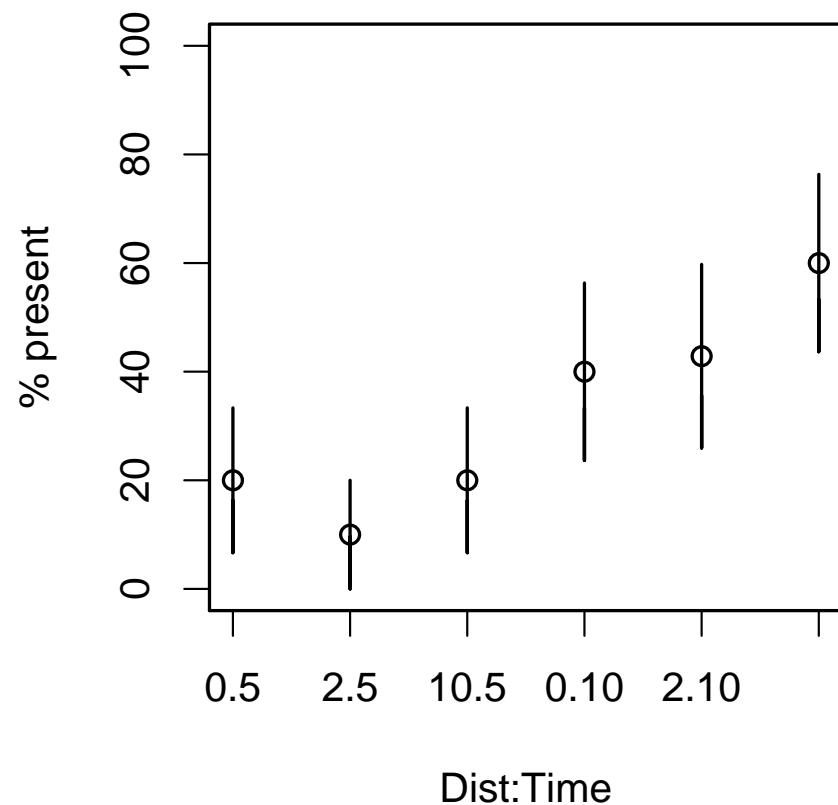


Using “jittering” i.e. add random noise to remove overlap:



N.B. Check out the `jitter` function in R

Alistair's mean-variance relationship (bars near 0% are shorter):



GLMs – relaxing linear modelling assumptions

Recall that linear models make the following assumptions:

1. The observed y values are **independent**, conditional on x
2. The y values are **normally distributed with constant variance**

$$y \sim \mathcal{N}(\mu_y, \sigma^2)$$

3. **straight line relationship** between mean of y and each x

$$\mu_y = \beta_0 + \mathbf{x}^T \boldsymbol{\beta}$$

Generalised linear model

Generalised linear models (GLMs) extend linear models to non-normal data. A GLM makes the following assumptions:

1. The observed y values are **independent**, conditional on x
2. The y values come from **a known distribution** (from the exponential family) with known **mean-variance relationship** $V(\mu)$
3. straight line relationship between **some known function of the mean** of y and each x

$$g(\mu_y) = \beta_0 + \mathbf{x}^T \boldsymbol{\beta}$$

The function $g(\cdot)$ is known as **the link function**.

Basically, a GLM adds two features to linear models:

- a **mean-variance relationship** $V(\mu)$, in place of constant variance
- a **link function** $g(\cdot)$ used to transform the mean before assuming linearity

What distributions can I use?

Not all distributions can be used with generalised linear models, but a few important ones can:

binomial for presence/absence data, or “ x -out-of- n ” counts across n independent events.

Poisson this should be your “default” for counts

negative binomial well, this is kind of a GLM. Great for count data which is too variable to fit a Poisson.

Tweedie again, kind of a GLM. Pretty good for modelling biomass data (Foster and Bravington, 2013).

normal In the special case where you assume y is normally distributed, GLM's reduce to linear models.

There are more distributions that you could use...

Mean-variance relationship

Each of these distributions assumes a special mean-variance relationship for the response variable:

binomial $V(\mu) = n\mu(1 - \mu)$. For presence-absence data, $n = 1$ so this simplified to $V(\mu) = \mu(1 - \mu)$

Poisson $V(\mu) = \mu$. Watch out for this assumption, it can be quite restrictive.

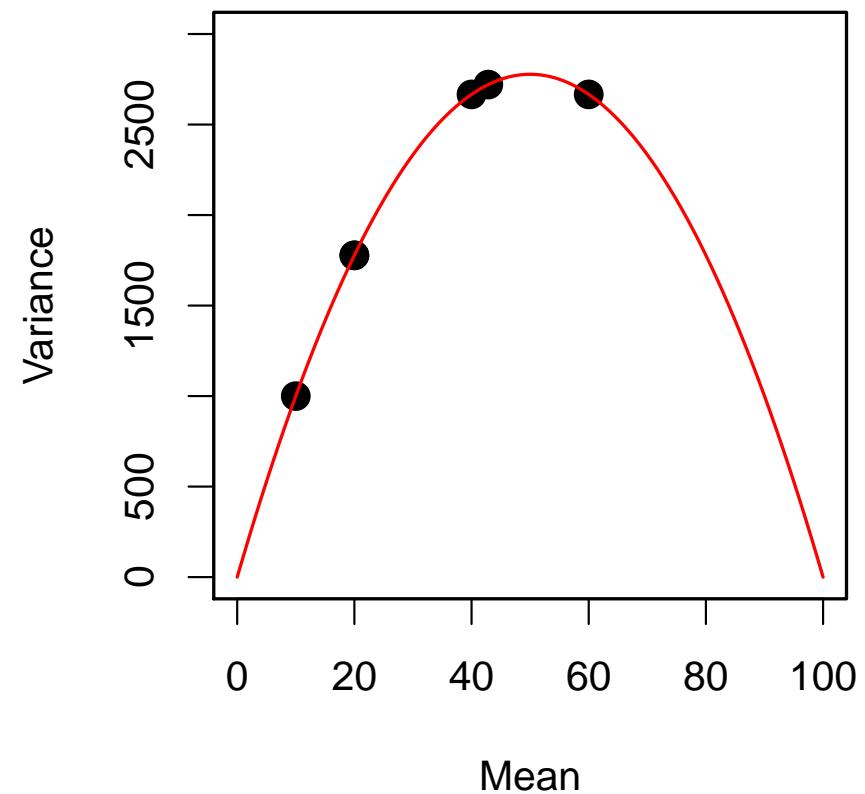
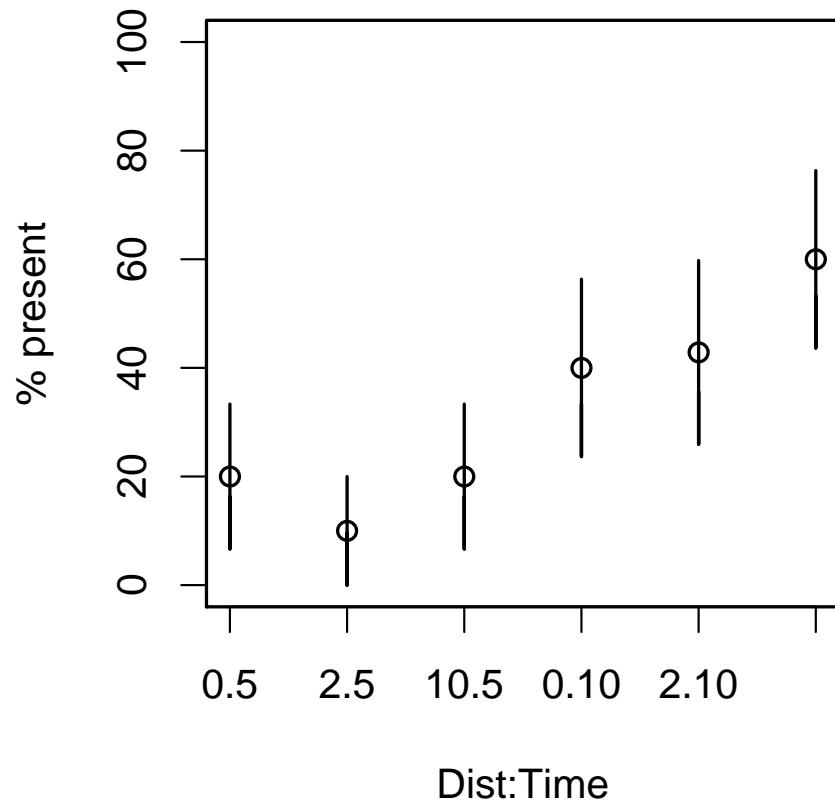
negative binomial $V(\mu) = \mu + \phi\mu^2$. There is a parameter in there (ϕ , the “overdispersion parameter”) which controls the degree to which the data are more variable than a Poisson (Usually $0 \leq \phi < 1$, if $\phi = 0$ then data are Poisson)

Tweedie $V(\mu) = a\mu^p$. This is cool because this form of mean-variance relationship is known as “Taylor’s power law”, and much ecological data seems to (approximately) follow it.

normal $V(\mu) = \sigma^2$. Note this is not a function of μ – that is, the normal distribution assumes the variance is a constant.

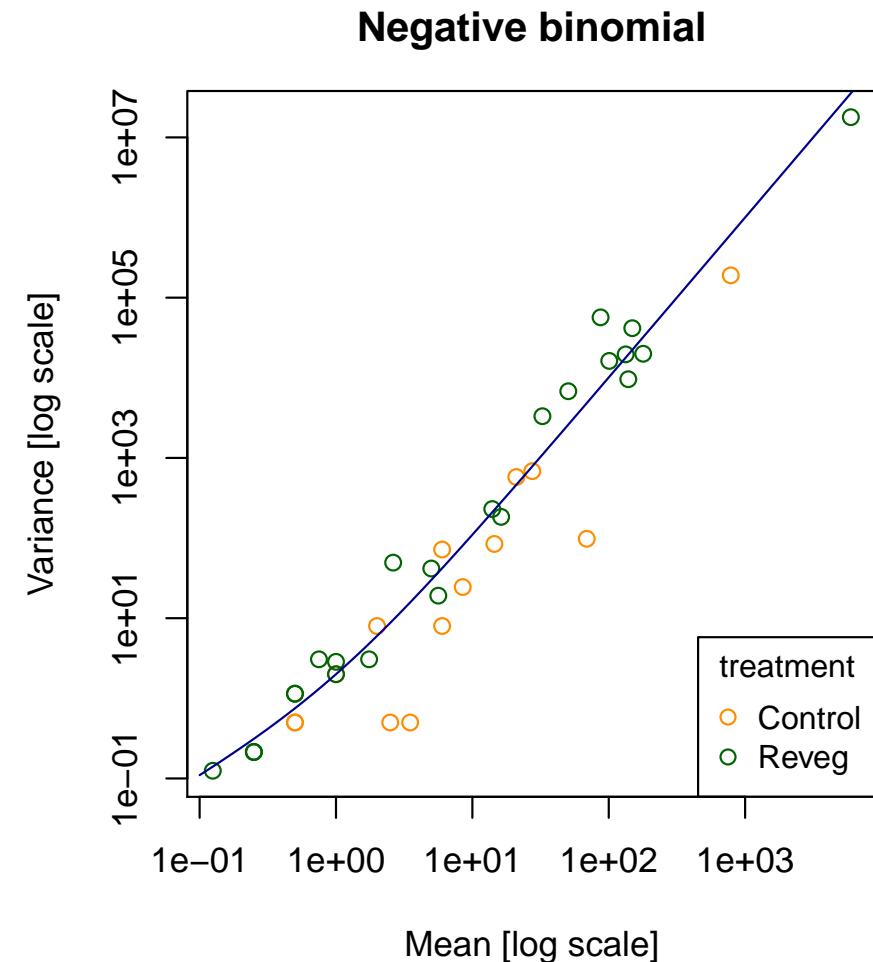
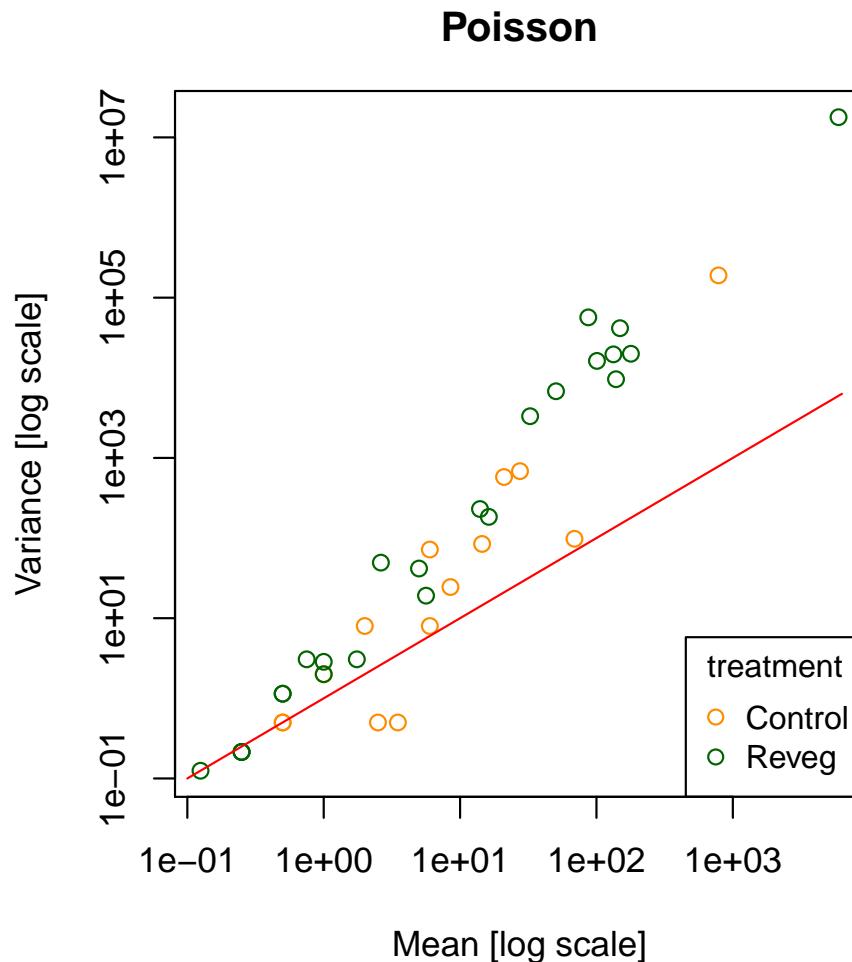
Alistair's mean-variance relationship – note that it is **exactly** quadratic

$$V(\mu) = n\mu(1 - \mu)$$



Mean-variance assumptions for reveg counts.

Which mean-variance assumption looks more plausible?



Link functions

Below are the common link functions used for different distributions.

binomial the logit function $\text{logit}(\mu) = \frac{\mu}{1-\mu} = \beta_0 + \mathbf{x}^T \boldsymbol{\beta}$. This is multiplicative in terms of the “odds”.

Sometimes the probit function $\Phi^{-1}(\mu)$ where Φ is the probability function of the standard normal.

Sometimes the complementary log-log link, $\log(-\log(1-\mu))$ (if you had Poisson log-linear counts which you truncated to pres/abs).

Poisson the log-link, $\log(\mu) = \beta_0 + \mathbf{x}^T \boldsymbol{\beta}$, is almost always used. This gives us a multiplicative model, often called a “log-linear model”.

negative binomial usually the log-link, $\log(\mu) = \beta_0 + \mathbf{x}^T \boldsymbol{\beta}$.

Tweedie usually the log-link, $\log(\mu) = \beta_0 + \mathbf{x}^T \boldsymbol{\beta}$.

normal usually the identity link, $\mu = \beta_0 + \mathbf{x}^T \boldsymbol{\beta}$.

In principle you can use any link function with any distribution. But in practice, you almost always should use a function that matches the domain of the mean.

e.g. Poisson means are always positive, so use a link function which works for positive values only, and maps them onto the whole number line. e.g. the log function.

On R, the default link function for the binomial is the logit-link (often referred to as **logistic regression**). For other distributions the default is the link given on the previous slide.

Notation warning – NOT a general linear model

Some refer to the linear model as a “general linear model”, GLM for short. *i.e.* a model where we assume the response is normally distributed, no mean-variance relationship, no link function. That is bad (and confusing) terminology and we can blame the SAS program – it is mostly their fault.

When people talk about GLMs make sure you are clear what they are talking about – do they mean a generalised linear model (non-normal response, mean-variance relationship...) or are they just using an simple linear regression and trying to sound technical?

Fitting and checking GLMs

You use the `glm` function:

```
> dathabconf = read.csv("HabitatConfig.csv", header = TRUE)
> dathabconf$CrabPres = dathabconf$Crab>0
> dathabconf$Time = factor(dathabconf$Time)
> dathabconf$Dist = factor(dathabconf$Dist)
> ft.crab = glm(CrabPres~Time*Dist, family=binomial, data=dathabconf)
> summary(ft.crab)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.39e+00	7.91e-01	-1.75	0.08 .
Time10	9.81e-01	1.02e+00	0.96	0.34
Dist2	-8.11e-01	1.32e+00	-0.62	0.54
Dist10	8.26e-16	1.12e+00	0.00	1.00
Time10:Dist2	9.29e-01	1.65e+00	0.56	0.57
Time10:Dist10	8.11e-01	1.44e+00	0.56	0.57

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

(Dispersion parameter for binomial family taken to be 1)

```
Null deviance: 71.097 on 56 degrees of freedom
Residual deviance: 62.999 on 51 degrees of freedom
AIC: 75
```

Family argument

Use the family argument in the following ways:

logistic regression family=binomial

Poisson log-linear family=poisson

probit regression family=binomial(link="probit")

complementary log-log regression family=binomial(link="cloglog")

Poisson linear family=poisson(link="identity")

linear model family=gaussian ("Gaussian" is another word for "normal")

For the negative binomial and tweedie distributions, you need to use a special package.

Remember to set the family argument! If you forget it, `glm` defaults to `family=gaussian` (a linear model).

How is a GLM fitted?

Maximum likelihood.

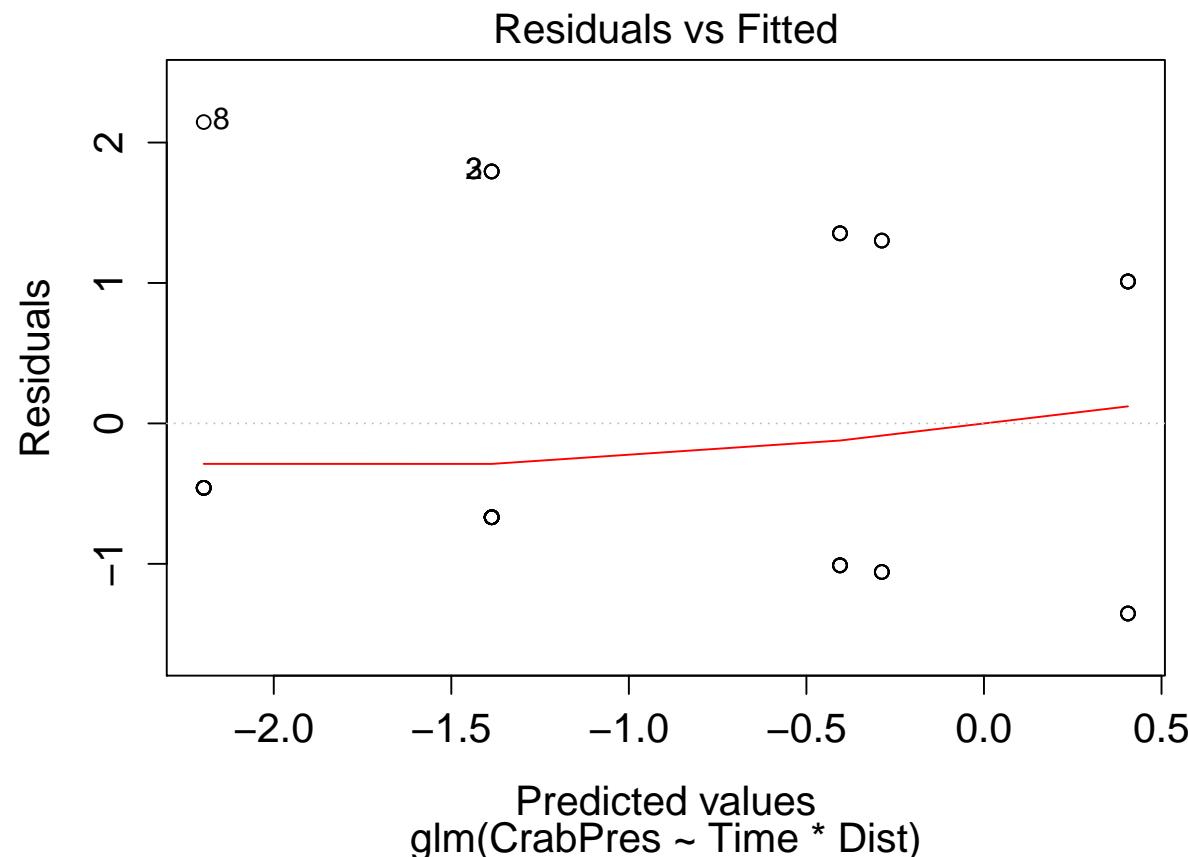
When measuring goodness-of-fit, we no longer talk about residual sums of squares (because we are not fitting the model by least squares). Instead we talk about **deviance** – basically, twice the difference in likelihood between the fitted model and a perfect model (predicted values being exactly observed values).

It is often said that a good model will have residual deviance similar in size to the residual degrees of freedom – this is a VERY rough rule though and it won't work if you have lots of zeros/small counts.

Does the residual deviance suggest that the fit to Alistair's crab data is any good?

How do you check assumptions?

Don't just look at numerical measures (residual deviance, AIC, etc)
– plot your data! As with linear models, use residual plots to checking
for no pattern. Try `plot(ft.crab)`:



How do you check assumptions?

The plot above looks really weird for a couple of reasons. The main problem is the discreteness – there are two lines of points on the plot (for zeros and for ones), which is a pattern that distracts us from the main game.

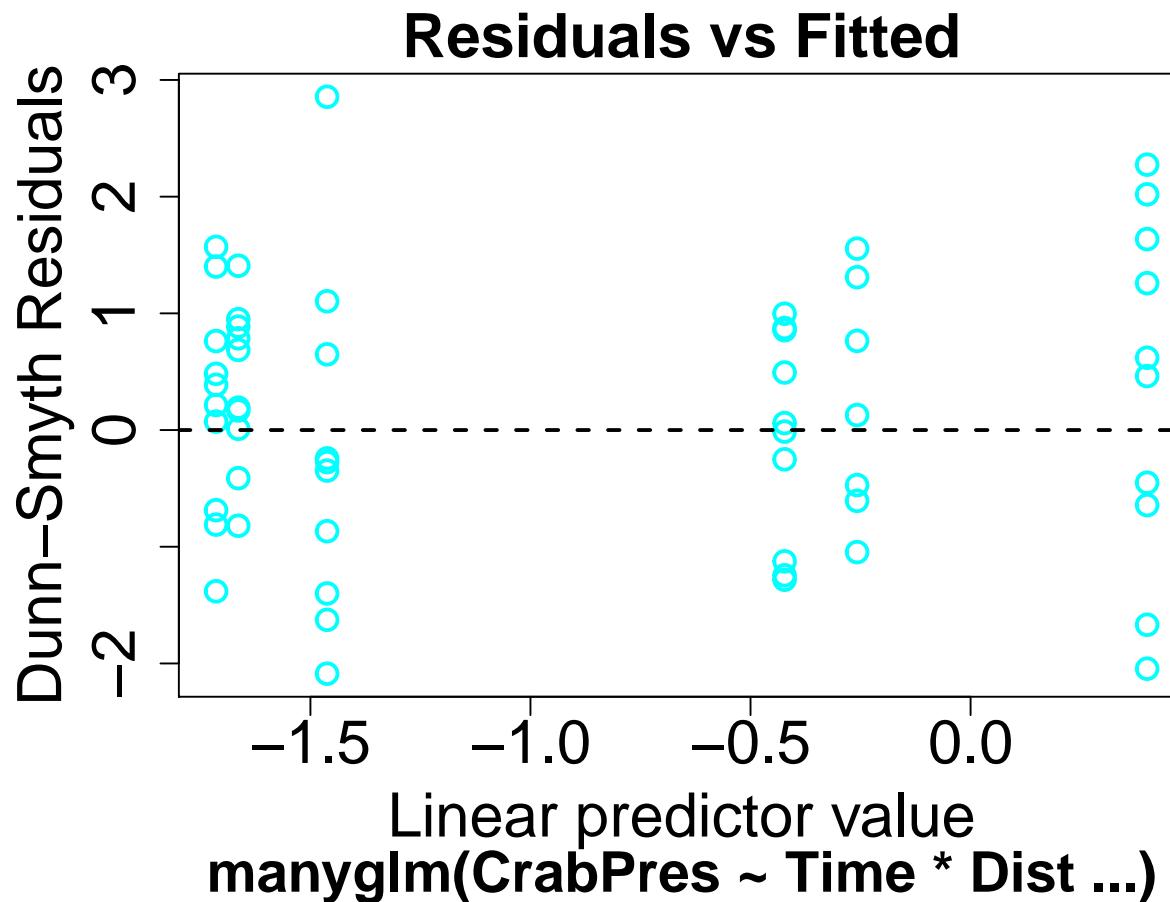
The problem is the idea of a residual – it is not obvious how to define residuals appropriately for GLMs. Most software (including R) chooses badly.

Dunn-Smyth residuals

One solution is to use residuals defined via the probability integral transform, which we refer to as **Dunn-Smyth residuals**. These residuals have been around for some time (Dunn & Smyth 1996) but their awesomeness is only starting to be really appreciated.

These residuals are not in standard packages (yet) but you can use them via the `mvabund` package by refitting your model using `manyglm`.

```
> library(mvabund)
> ft.crab2 = manyglm(CrabPres~Time*Dist, family="binomial", data=dathabconf)
> plot(ft.crab2)
```



Aside: Small change to family argument for mvabund

If you look carefully in the previous command you will notice we used
family="binomial"

not

family=binomial

Quotation marks are currently required by the `manyglm` function for all families but we are hoping to change that...

Interpret residual plots like for linear models

Recall that for linear regression:

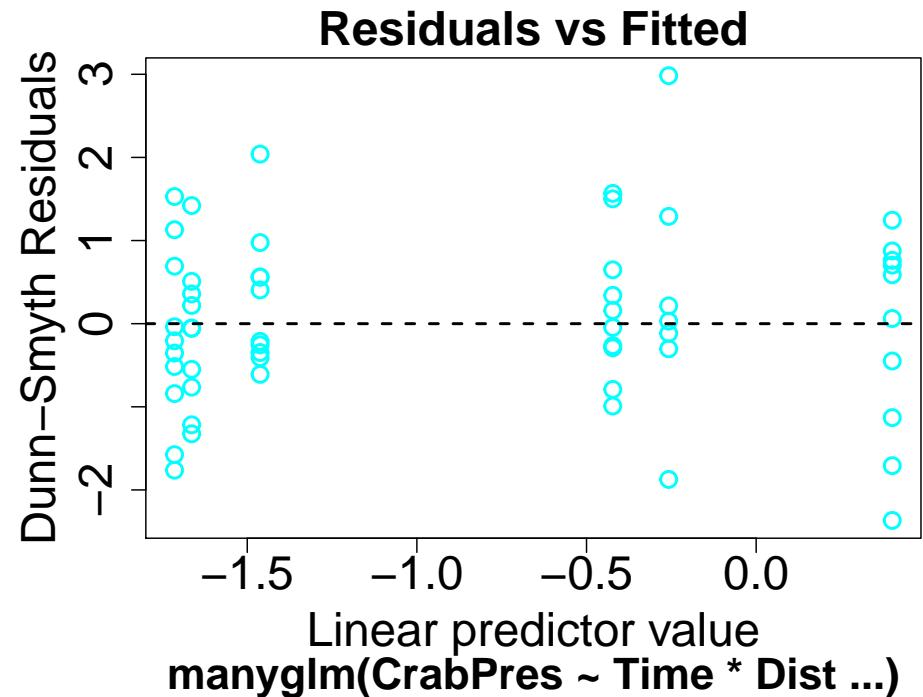
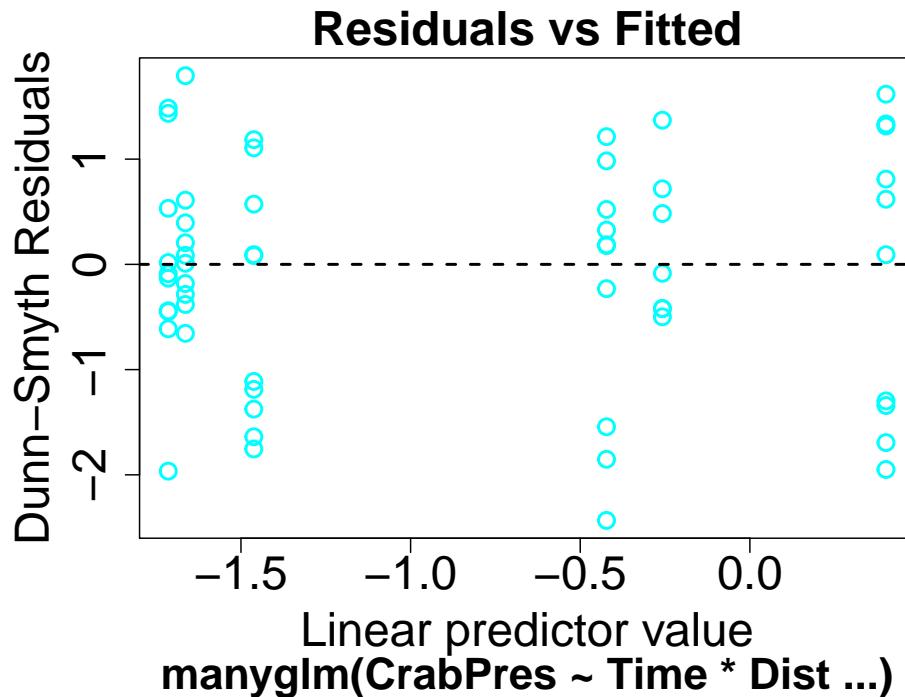
- U-shape \Rightarrow violation of straight line assumption
- Fan-shape \Rightarrow violation of variance assumption

Dunn-Smyth plots work the same way:

- U-shape violation of linearity assumption
- Fan-shape violation of mean-variance assumption

Dunn-Smyth residuals deal with the problem of discreteness through random number generation – basically, jittering points.

This means different plots will give you different residuals...



Plot residuals more than once to check if any pattern is “real” .

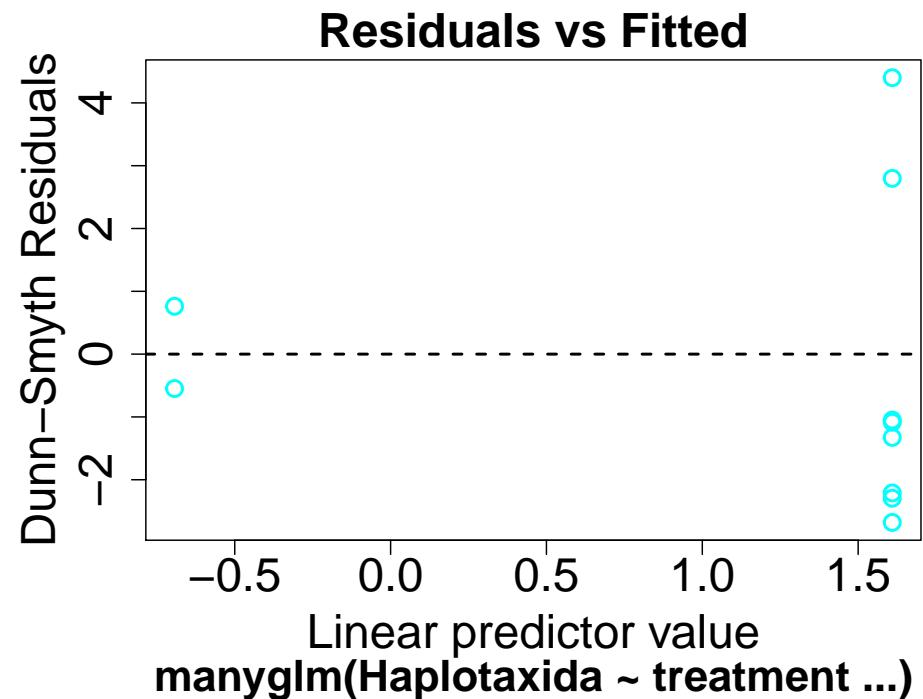
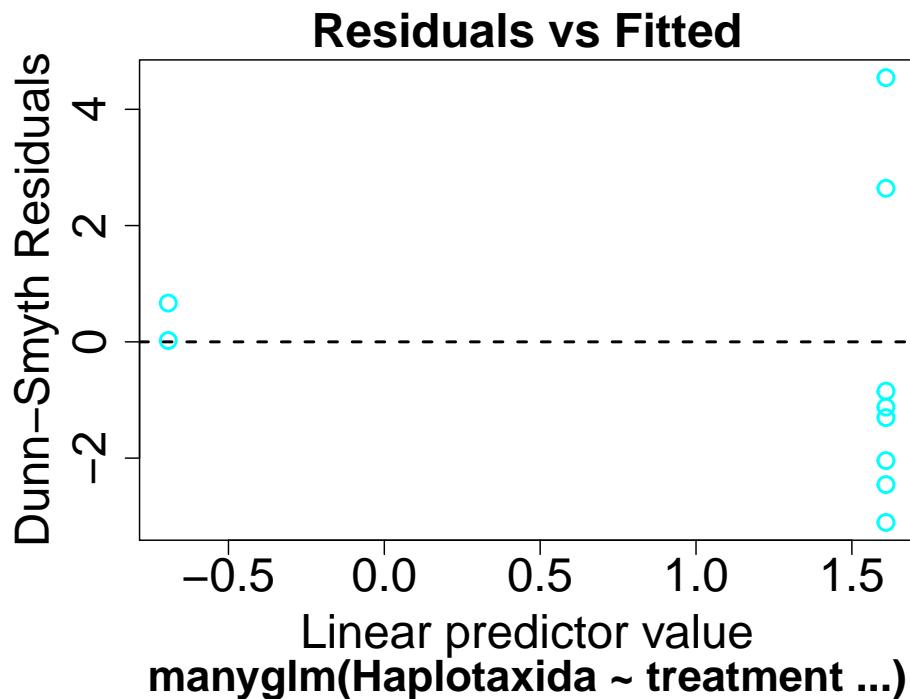
Counts that don't fit a Poisson distribution

Recall that Anthony's revegetation data looked like it would be better fitted by a negative binomial distribution (Slide 6.17). Further evidence is seen for Haplotauxida:

```
> datRed = read.csv("revegSmall.csv")
> ft.hap = manyglm(Haplotauxida~treatment,family="poisson", data=datRed)
> plot(ft.hap)
```

Counts that don't fit a Poisson distribution

Is there any consistent pattern?



Counts that don't fit a Poisson distribution

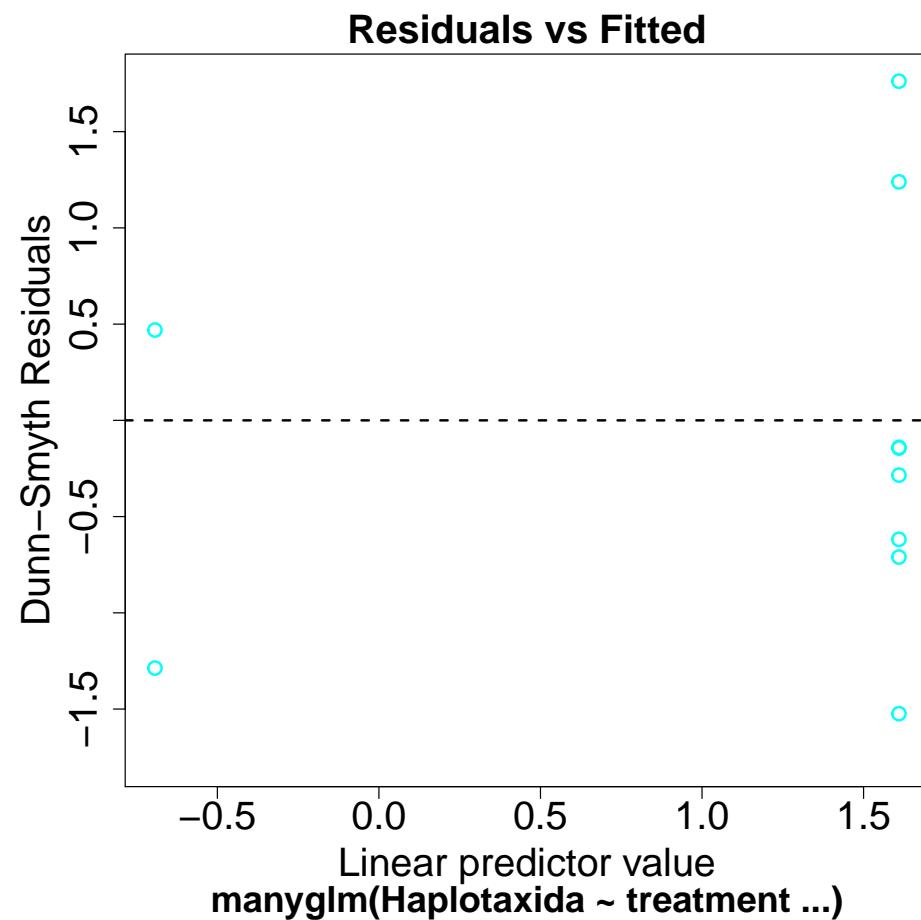
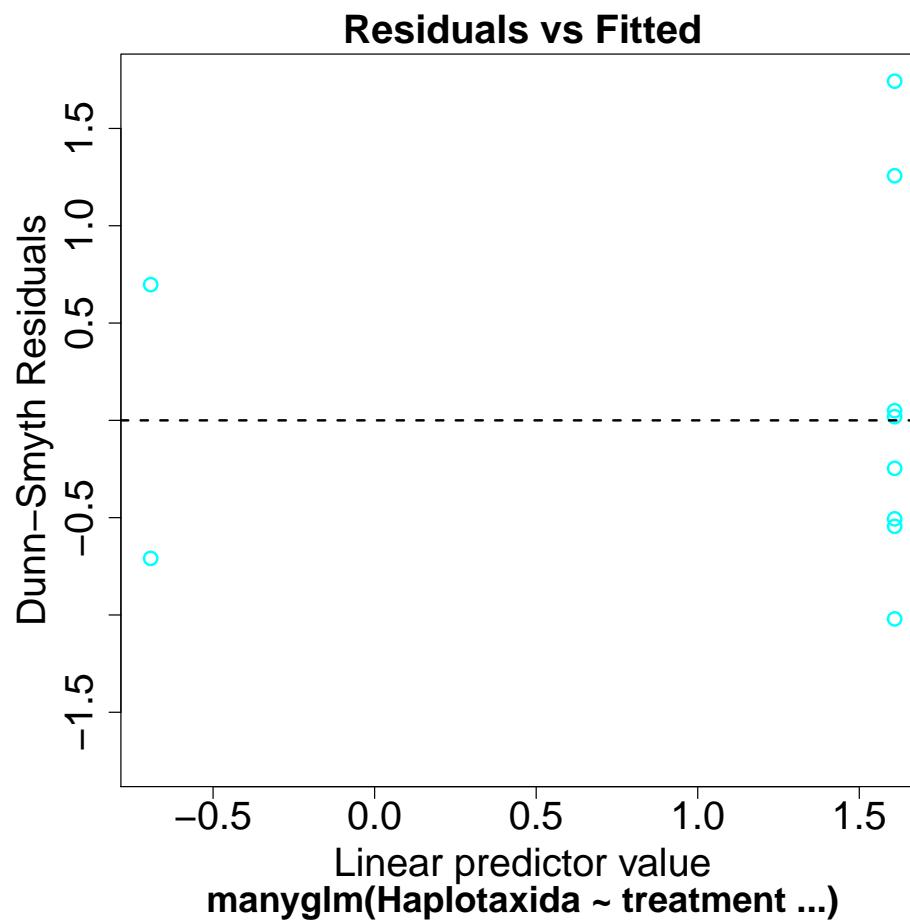
At larger values of the mean, data are more variable than expected. This is because the Poisson mean variance assumption ($V(\mu) = \mu$) can be a bit restrictive. Instead we will use the negative binomial.

Negative binomial regression can be easily fitted using the `mvabund` package

```
> ft.hap2 = manyglm(Haplotypeida~treatment, family="negative.binomial", data=datRed)
> plot(ft.hap2)
```

Counts that don't fit a Poisson distribution

Is the negative binomial distribution producing a better model?



Inference from generalised linear models

All the same functions as for linear models work:

Confidence intervals use the `confint` function

Hypothesis testing use the `summary` or `anova` function. The latter is generally better for hypothesis testing.

Model selection works as previously (`stepAIC`, `predict` for cross-validation)

Using the `anova` function with GLMs

The term `anova` is a little misleading for GLMs – technically, what we get is an analysis of deviance table, not analysis of variance.

For GLMs we have to tell `anova` what test statistic to use (otherwise it won't use any!). Add the argument `test="Chisq"` – it will do likelihood ratio tests, and compare test statistics to a chi-squared distribution to get *P*-values.

Using the anova function with GLM

```
> ft.crab = glm(CrabPres~Time*Dist, family=binomial, data=databconf)
> anova(ft.crab, test="Chisq")
```

...

Terms added sequentially (first to last)

	Df	Deviance	Resid.	Df	Resid.	Dev	Pr(>Chi)
NULL				56		71.1	
Time	1	6.67		55	64.4	0.0098	**
Dist	2	0.99		53	63.4	0.6085	
Time:Dist	2	0.43		51	63.0	0.8048	

...

Is there any evidence of an effect of Isolation distance or Time on crabs?

summary vs anova for GLMs

The `summary` function on the other hand uses Wald tests – comparing $\hat{\beta}/\text{se}(\hat{\beta})$ to a standard normal distribution. This is less accurate, and especially for logistic regression, can give quite different results:

```
> summary(ft.crab)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.39e+00	7.91e-01	-1.75	0.08 .
Time10	9.81e-01	1.02e+00	0.96	0.34
Dist2	-8.11e-01	1.32e+00	-0.62	0.54
Dist10	8.26e-16	1.12e+00	0.00	1.00
Time10:Dist2	9.29e-01	1.65e+00	0.56	0.57
Time10:Dist10	8.11e-01	1.44e+00	0.56	0.57

Are there any differences in results, as compared to the `anova` output previous?

Inference for small samples

The `summary` and `anova` tests are both approximate – they work well in large samples (well, `summary` can be a bit weird) but can be quite approximate when sample size is small.

We can beat this problem exactly by calculating P -values by simulation, specifically by resampling the data.

The simplest way to do this is using the `mvabund` package, which uses resampling by default whenever you call `summary` or `anova` for a `manyglm` object.

Inference for small samples

```
> ftmany.Hap=manyglm(Haplotaxida~treatment, family="negative.binomial", data = datRed)
> anova(ftmany.Hap)
Time elapsed: 0 hr 0 min 0 sec
Analysis of Deviance Table

Model: manyglm(formula = Haplotaxida ~ treatment, family = "negative.binomial",
Model:       data = datRed)

Multivariate test:
             Res.Df Df.diff   Dev Pr(>Dev)
(Intercept)      9
treatment        8       1 2.81     0.16
Arguments: P-value calculated using 999 resampling iterations via PIT-trap resampling.
```

Is there any evidence of an effect of revegetation on worm counts?

Inference for small samples

```
> ftmany.crab = manyglm(CrabPres~Time*Dist, family="binomial", data=dathabconf)
> anova(ftmany.crab)
Time elapsed: 0 hr 0 min 0 sec
Analysis of Deviance Table

Model: manyglm(formula = CrabPres ~ Time * Dist, family = "binomial",
Model:       data = dathabconf)

Multivariate test:
             Res.Df Df.diff   Dev Pr(>Dev)
(Intercept)    56
Time          55      1  6.67    0.02 *
Dist          53      2  0.99    0.65
Time:Dist     51      2  0.43    0.88
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
Arguments: P-value calculated using 999 resampling iterations via PIT-trap resampling.
```

Are there any important differences from results previously obtained?

Options for resampling methods in `mvabund`

The default resampling method in the `mvabund` package is the PIT-trap – a residual resampling method which bootstraps residuals computed via the probability integral transform. Basically, it bootstraps Dunn-Smyth residuals. This is something we have recently developed in the Eco-Stats Research group, and so far it works better in simulations than anything else we've tried.

Currently, the most popular method available for resampling GLMs is the parametric bootstrap (more on this later in the mixed models session). You can use the parametric bootstrap in `mvabund` by setting `resamp="monte.carlo"`, e.g.

```
ftmany.crabMC = manyglm(CrabPres~Time*Dist, family="binomial",
data=dathabconf, resamp="monte.carlo")
```

Extensions of GLMs

There are a few important additional features and extensions of GLMs worth knowing about.

Offsets

Sometimes there is some variable known not just to be important to the response, but its precise relationship with response is also known. This most commonly happens with sampling intensity (sample twice as hard you expect to get twice as many observations).

e.g. Anthony actually sampled five pitfall traps in nine sites, but only four pitfall traps in a tenth site, as follows:

Treatment	C	R	R	R	C	R	R	R	R	R
Count	0	3	1	3	1	2	12	1	18	0
# pitfalls	5	5	5	5	5	5	5	4	5	5

How can we account for the different sampling effort at different sites in our model?

Offsets

Don't just rescale or average values. This changes the distribution of your data, and stuffs up the mean-variance relationship.

Instead, we include an **offset** – a predictor variable known to be exactly proportional to the response. Because we model $\log(\mu)$ in Poisson and negative binomial regression, our offset is $\log(\# \text{ pitfalls})$.

This is most easily done by adding an offset of `log(pitfalls)` to your model formula using:

```
offset(log(pitfalls))
```

Offsets

```
> ftmany.hapoffset = manyglm(Haplotaxida~treatment+offset(log(pitfalls)),  
  family="negative.binomial", data = datRed)  
> anova(ftmany.hapoffset)  
Time elapsed: 0 hr 0 min 0 sec  
Analysis of Deviance Table
```

Model: manyglm(formula = Haplotaxida ~ treatment + offset(log(pitfalls)),
Model: family = "negative.binomial", data = datRed)

Multivariate test:

	Res.Df	Df.diff	Dev	Pr(>Dev)
(Intercept)	9			
treatment	8	1	3.14	0.14

Arguments: P-value calculated using 999 resampling iterations via PIT-trap resampling.

Why didn't we hear about offsets for linear models?

Because linear models are additive, so you could just subtract the offset from y before fitting the model.

That is, in a linear model for $\log(\text{Haplotaxida})$, to include an offset for $\log(\text{pitfalls})$ you would model

$$\log(\text{Haplotaxida}) - \log(\text{pitfalls})$$

as a function of treatment.

Zero-inflated models

Ecological counts often have many zeros e.g. consider Anthony's revegetation counts:

Treatment	C	R	R	R	C	R	R	R	R	R
Count	3	0	0	0	4	1	0	0	0	0
# pitfalls	5	5	5	5	5	5	5	4	5	5

It is tempting to use a **zero-inflated** model (Welsh et al. 1996) – a model which expects more zeros than the Poisson.

For details on how to fit such models, see the `VGAM` and `pscl` packages.

However – just because you have lots of zeros doesn't mean that your data are necessarily zero-inflated. (e.g. a Poisson distribution with $\mu = 0.1$ already expects 90% of values to be zero!)

The above cockroach data are actually very well fitted by a Poisson distribution.

If trying out a zero-inflated model, please **check that you needed it** – maybe your data aren't actually zero-inflated (Warton, 2005)!

Generalised additive models

Sometimes the assumption of straight-line relationship $\mathbf{x}^T \boldsymbol{\beta}$ is too restrictive. One way to extend this is replace it with **smoothers** e.g. spline smoothers.

Such models are referred to as Generalised additive models (GAMs). For more details, see Faraway (2005). In R, these may be fitted using the `mgcv` package. Note however that the `mgcv` package does not use Dunn-Smyth residuals for residual plots.

N.B. It is important to remember that a “generalised linear model” does **not** need to be linear: by including functions of x as predictors (e.g. quadratic, cubic terms, periodic functions), you can use the “straight line relationship” to fit some fairly non-linear functions.

Generalised linear mixed models

Arguably the most important extension to GLMs is the inclusion of **random effects**, a topic that we shall cover in the next...

Sometimes you might have random factors (e.g. nested design). Generalised linear models (the `glm` and `manyglm` functions) only handle fixed effects.

If you have random effects and a non-constant assumed mean-variance relationship then you want to fit a **generalised linear mixed model**.

A good package for this, as in the linear mixed effects case, is the `lme4` package.

There aren't really any new tricks – just use the `glmer` argument as you would normally use `lmer`, but be sure to add a `family` argument.

Current limitations:

- As before, no nice residual plots (on our to-do list).
- As before, doesn't handle `family=negative.binomial` with unknown overdispersion. But you can try a workaround where you use the Poisson with additional random effects to introduce overdispersion.
- GLMM's can take much longer to fit and even then only give approximate answers. The mathematics of GLMM's are way harder than anything else we've seen this week.
- There is an optional argument `nAGQ` that you can try for simple models to get a better approximation for GLMMs (e.g. `nAGQ=4`).

Multivariate data

- Introduction to multivariate data
- MANOVA and multivariate linear models
- Visualising multivariate data

Introduction to multivariate data

multivariate = many variables.

Recall that the type of regression model you use is determined by the properties of the **response variable**. If you have multiple response variables, and you want to make inferences simultaneously across all of them, you should fit a multivariate model.

Examples

Ian likes studying leaves. He is especially interested in “leaf economics” – things like construction costs per unit area (`lma`), how long-lived leaves are (`longev`) and how they vary with environment. In particular:

Is there evidence that leaves vary (in their “economics” traits) across sites with different levels of rainfall and soil nutrients?

Because Ian is interested in leaf economics traits collectively, he has more than one response variable – he would like to look at `lma` and `longev` (and any other relevant traits) jointly.

Edgar was studying Iris flowers, and measured four size variables on each of 50 flowers from each species

What are the main sources of variation across the three species of Iris?

Are there any key characteristics of flowers that differentiate the three species?

Anthony classified anything that fell into his pitfall traps to Order, and is interested in using his data to answer the question:

Is there evidence of a change in invertebrate communities due to bush regeneration efforts?

He has a single question he wants to answer, using abundances from 24 orders of invertebrate as his response. This is multivariate! (very)

Challenges of multivariate analysis

Data visualisation We no longer have one response variable. We can look at responses one-at-a-time (or two-at-a-time) but if there are any higher-order patterns in correlation we will have a hard time seeing them. Or checking correlation assumptions.

Interpretation More variables means there is a lot going on and it is hard work to get a simple answer. If there is one.

Assumption violations More response variables means more ways model assumptions can be violated. These violations have a cumulative effect – lots of variables with modest violations can have a big effect.

High-dimensional data If you have a comparable number of response variables as you have observations (as Anthony does), model-based multivariate inference is out of the question, at the moment.

First step – multivariate, really???

Everything is a lot harder in multivariate world. Do you really need to go there???

If you can get a satisfactory answer to your key research question without going multivariate then by all means do so!

One common simplification is to compute some univariate statistic to characterise your community (e.g. species richness, total abundance). If this works (in the sense that it answers the question you are interested in, with a reasonable level of precision) and you and your people are happy with that then you are done.

MANOVA and multivariate linear models

The multivariate generalisation of analysis of variance is known as MANOVA (Multivariate ANalysis Of VAriance). It is a special case of multivariate linear models.

A multivariate linear model makes the following assumptions:

1. The observed y values are **independent** (after conditioning on x)
2. The y values are **multivariate normally distributed with constant variance-covariance matrix**

$$\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

3. **straight line relationship** between mean of each y and each x

$$\mu_j = \beta_{0j} + \mathbf{x}^T \boldsymbol{\beta}_j$$

Variance-co-what?

A variance-covariance matrix is a square grid of values storing the variance of each variable (along a diagonal) and the covariances. Covariances are basically the correlations between each pair of variables, re-scaled to the original units of measurement.

The variance-covariance matrix is an important descriptor of multivariate data – it is a multivariate generalisation of the idea of variance, which also captures the idea of correlation.

In the univariate case we made an assumption of constant variance. In the multivariate case we make an assumption of constant variance-covariance matrix, which means we assume:

- Every response variable has constant variance
- The correlations between all pairs of response variables are constant

Fitting multivariate linear models on R

This is easy – the only thing you have to do differently to the univariate case is to make sure your response variables are stored together in a **matrix** (can be done using `cbind`).

```
> library(smatr)
> data(leaflife)
> Yleaf <- cbind(leaflife$lma, leaflife$longev)
> ft.leaf = lm(Yleaf~rain*soilp, data=leaflife)
> anova(ft.leaf, test="Wilks")
Analysis of Variance Table

          Df    Wilks approx F num Df den Df    Pr(>F)
(Intercept) 1 0.11107  248.096      2     62 < 2.2e-16 ***
rain         1 0.68723   14.108      2     62 8.917e-06 ***
soilp        1 0.93478    2.163      2     62   0.1236
rain:soilp   1 0.95093    1.600      2     62   0.2102
Residuals    63

---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Any evidence of an effect of rainfall or soil nutrients?

Different test statistics

There are different types of test statistic for multivariate linear modelling:

Wilk's lambda The likelihood ratio statistic, `anova(ft.leaf1, test="Wilks")`

Hotelling-Lawley `anova(ft.leaf1, test="Hotelling-Lawley")`

Pillai-Bartlett trace the default, `anova(ft.leaf1, test="Pillai")`

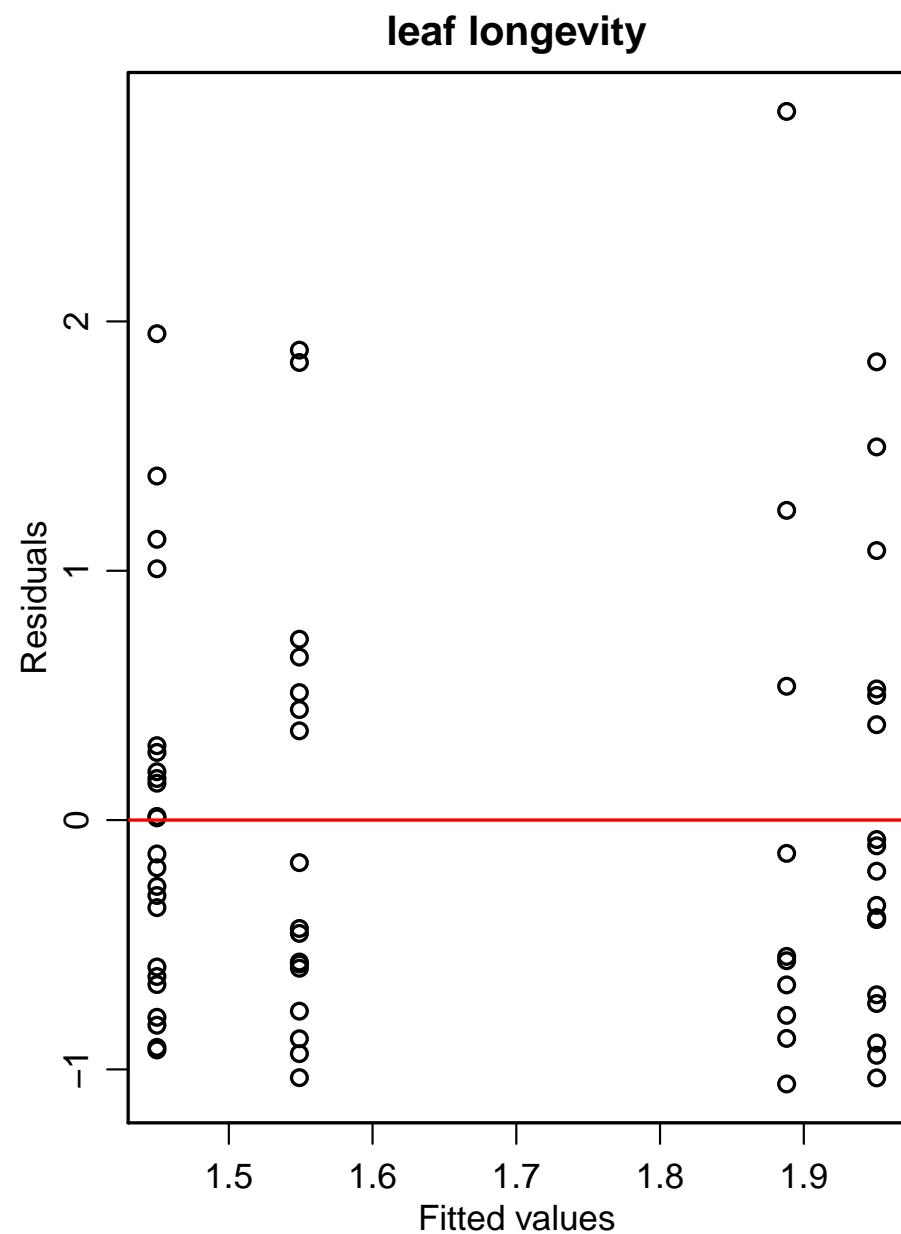
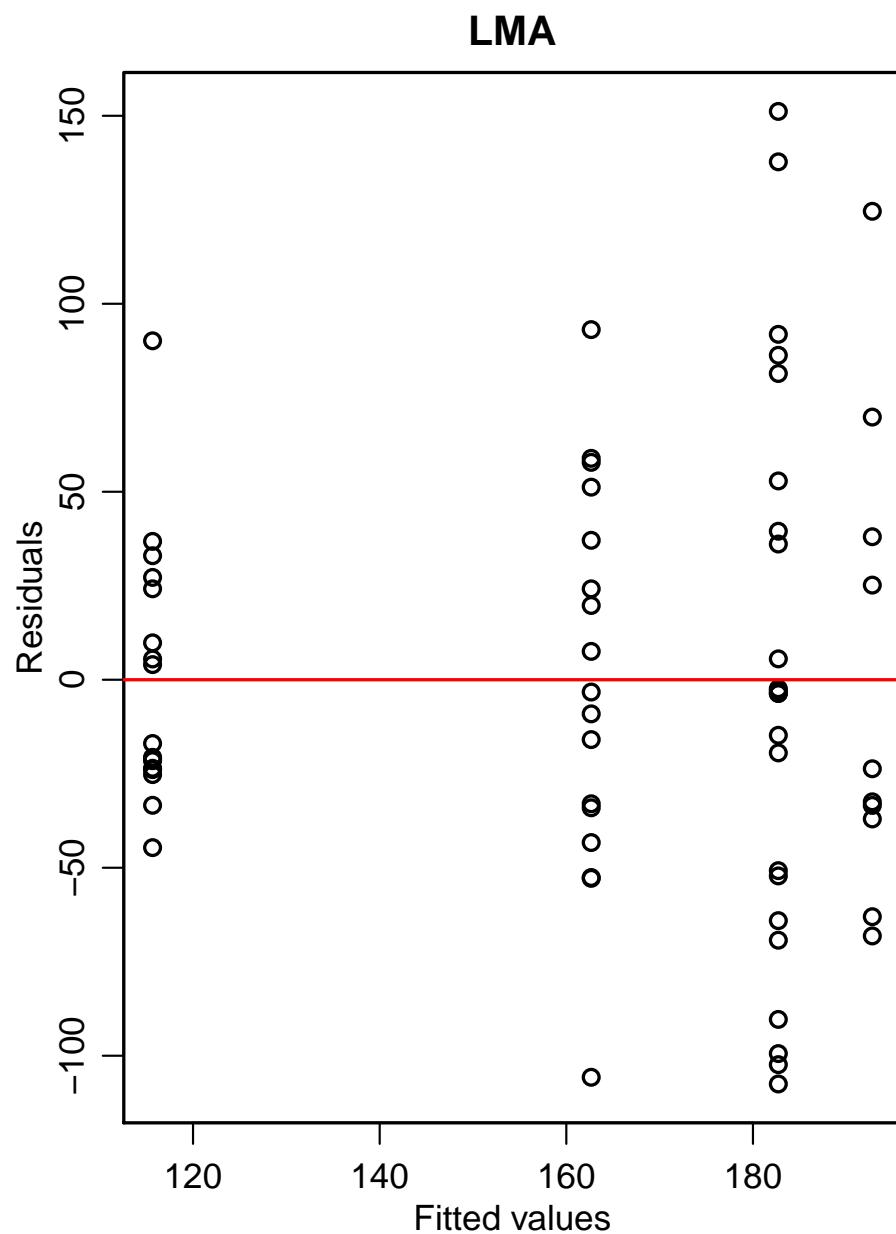
Roy's greatest root `anova(ft.leaf1, test="Roy")`

It is worthwhile using more than one of these to confirm a null result – each is good for slightly different things.

Checking assumptions

As usual, I recommend checking residual plots for no pattern. But the `plot` function doesn't work for multivariate linear models so there is a little work to do:

```
> par(mfrow=c(1,2))
> plot(residuals(ft.leaf)[,1]~fitted(ft.leaf)[,1], main="LMA",
       xlab="Fitted values", ylab="Residuals")
> abline(h=0,col="red")
> plot(residuals(ft.leaf)[,2]~fitted(ft.leaf)[,2],
       main="leaf longevity", xlab="Fitted values", ylab="Residuals")
> abline(h=0,col="red")
```



Multivariate linear models on `mvabund`

The `mvabund` package can also fit multivariate linear models, via the `manylm` function. The main differences:

- (as earlier today) it uses (residual) resampling, for design-based inference. **Rows** of residuals are resampled to ensure correlation is accounted for in testing.
- The default setting doesn't control for correlation, you have to use `cor.type="R"`.
- The test stats are called different things – Hotelling-Lawley is `test="F"`, and Wilk's lambda is `test="LR"` (Likelihood Ratio)

Using `cor.type="R", test="LR"` is equivalent to Wilk's test (even though the value of the test statistic is different).

```

> ft.leaf = lm(Yleaf~rain*soilp, data=leaflife)
> anova(ft.leaf, test="Wilks")
Analysis of Variance Table

```

	Df	Wilks	approx F	num Df	den Df	Pr(>F)
(Intercept)	1	0.11107	248.096	2	62	< 2.2e-16 ***
rain	1	0.68723	14.108	2	62	8.917e-06 ***
soilp	1	0.93478	2.163	2	62	0.1236
rain:soilp	1	0.95093	1.600	2	62	0.2102
Residuals	63					
...						

```

> ftmany.leaf = manylm(Yleaf~rain*soilp, data=leaflife)
> anova(ftmany.leaf, cor.type="R", test="LR")
Analysis of Variance Table

```

	Res.Df	Df.diff	val(LR)	Pr(>LR)
(Intercept)	66			
rain	65	1	23.941	0.001 ***
soilp	64	1	4.475	0.125
rain:soilp	63	1	3.371	0.220

What's the point of `manylm`?

If it gives similar answers to `lm`, what's the point?

The `anova` function for multivariate linear models is model-based, and uses large-sample approximations to get the P -value. This sometimes goes wrong, for two reasons:

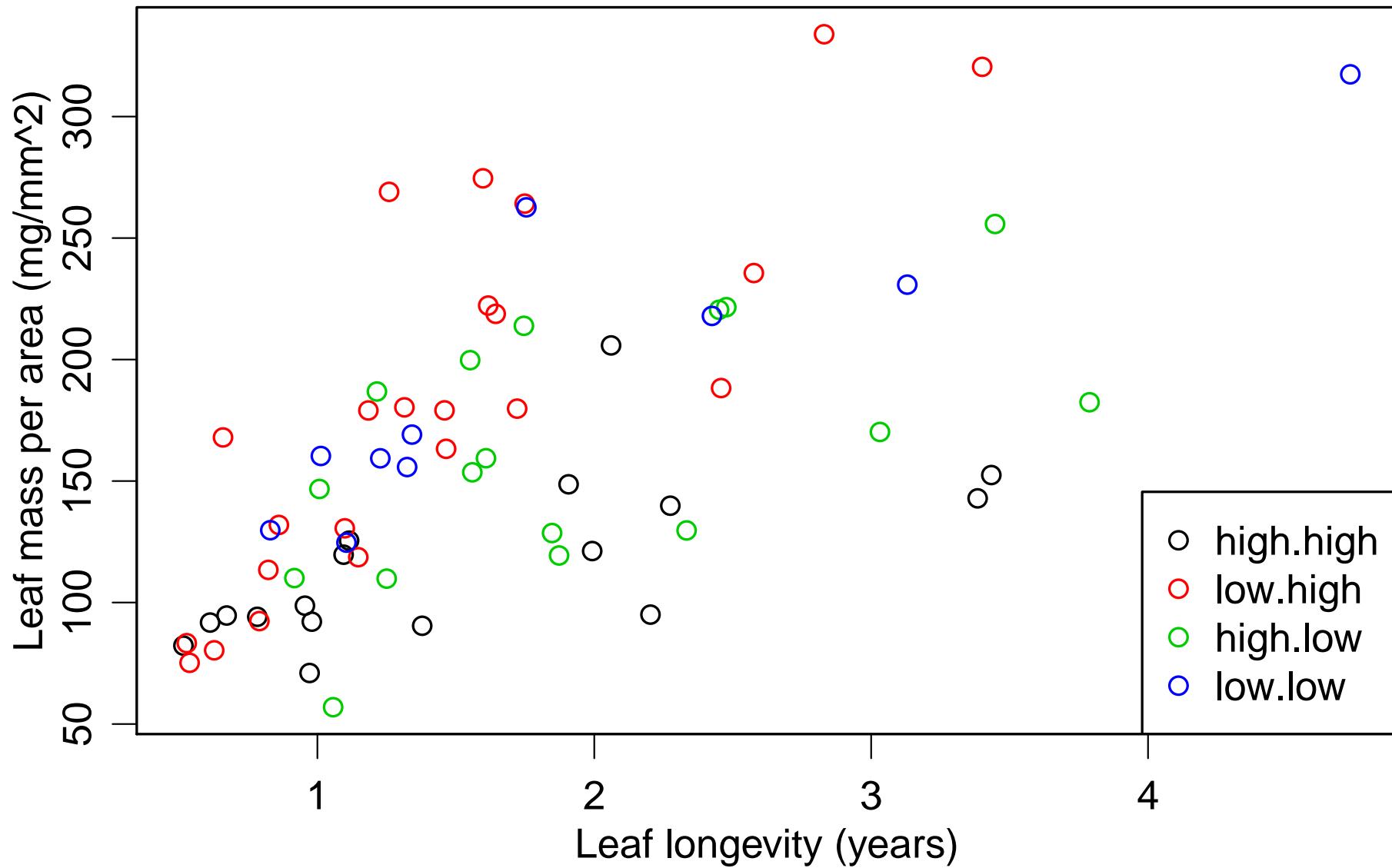
- If data are not normal (but other assumptions are not relaxed – just normality)
- If you have a decent number of variables compared to the number of observations (maybe more than a quarter?)

The second of these points is the big one. Testing via `lm` falls over when there are many variables, even when data are multivariate normal, but `manylm` keeps working in these cases. (But if there are many variables try `cor.type="shrink"` or `cor.type="I"` for a more efficient statistic.)

Visualising multivariate data

Ian is lucky because he only has two variables so data visualisation is a bit easier:

```
par(mfrow=c(1,1))
plot(leaflife$lma~leaflife$longev, xlab="Leaf longevity (years)",
      ylab="Leaf mass per area (mg/mm^2)",
      col=interaction(leaflife$rain,leaflife$soilp))
legend("bottomright",
      legend=levels(interaction(leaflife$rain,leaflife$soilp)),
      col=1:4, pch=1)
```



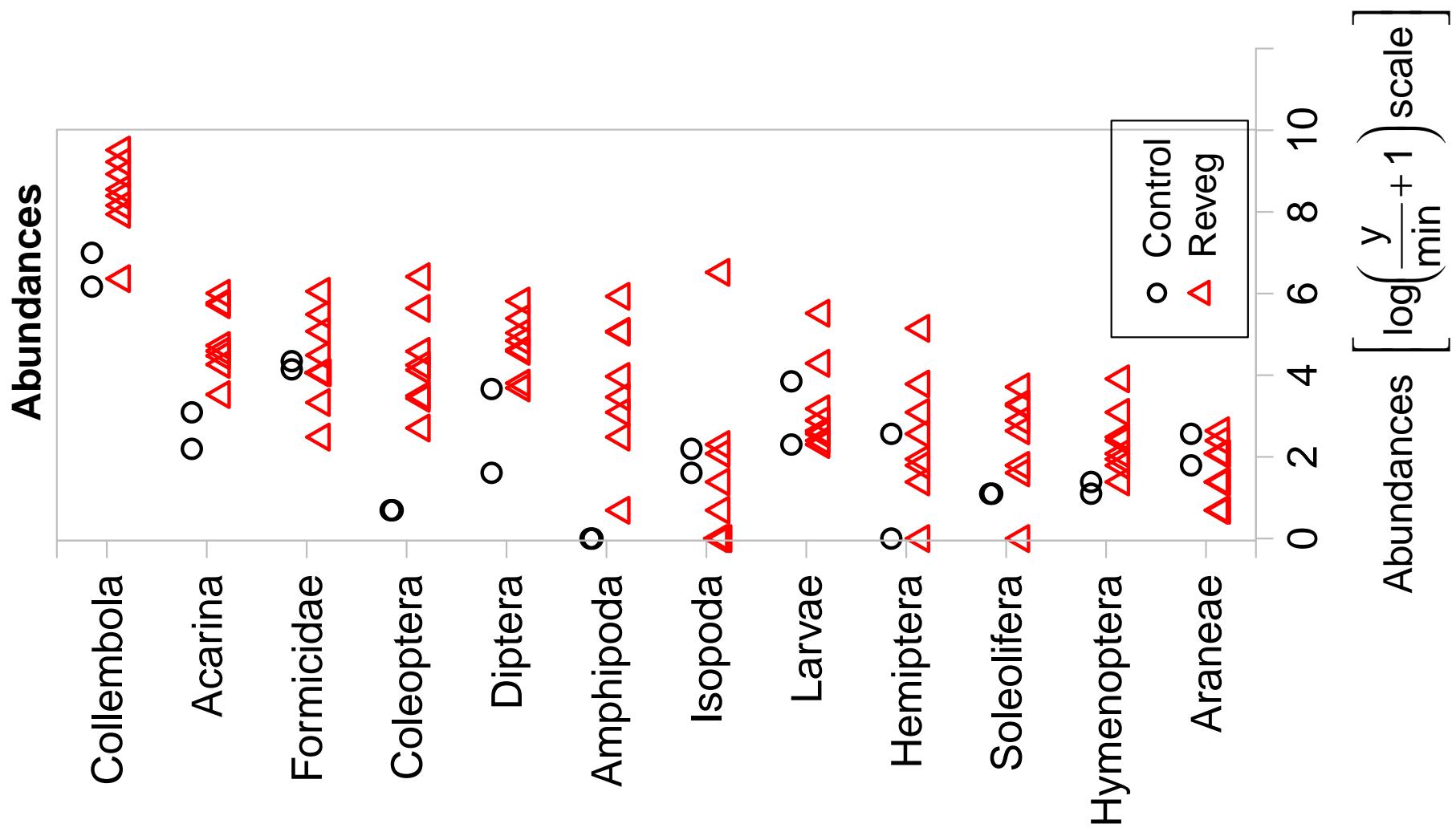
Notice Ian's leaf economic variables are **correlated**.

This raises the possibility that by looking at both variables jointly you can see structure that you might have missed from looking at each variable separately.

So what do you do if you have many response variables?

A good starting point is to plot each variable separately. The `mvabund` package has a `plot` function which makes this a bit easier to do. For Anthony's data:

```
> library(mvabund)
> load("reveg.RData")
> reveg2=mvabund(reveg) #so mvabund knows to treat dat as multivariate
> plot(reveg2~treatment)
```



Can you see any taxa that seem to be associated with bush regeneration?

plot.mvabund only plots a subset of variables

Note that by default this plot function only showed 12 variables – the 12 with highest total abundance. You can change `n.vars` to control how many taxa are plotted, and `var.subset` to control which taxa are plotted.

Visualising correlation between response variables

You can learn a fair bit about the relationship between y and x from those types of plots (separate plots of y against x for each y variable). But this just plots the “marginal effect” on y – if there is a more complex relationship between x and combinations of y , you can’t see it without jointly plotting y variables against x in some way.

One option is a scatter plot matrix with colour-coded x variables (e.g. pairs). But if there are too many variables this is a nightmare (e.g. for 100 variables there are **4950** possible pairwise scatterplots!)

Another option is **ordination**.

Principal components analysis

Principal components analysis (PCA) is the simplest ordination tool around. It uses either the variance-covariance matrix or the correlation matrix to find a rotation of the data to explain as much variation as possible. You can use the `princomp` function.

If variables are measured on the same scale you can do a PCA using the covariance matrix (default), but if large differences in variation or measured using different scales it would make sense to standardised (`cor=TRUE`).

Being based on the variances and correlations, PCA is sensitive to outliers and skewed variables – you want data to be fairly symmetric and not too far from normally distributed. For data with outliers you could do a PCA on covariances as estimated using robust methods, e.g. using `cov.rob` from the `MASS` package.

A good option for morphological data – a bad option for multivariate abundance data.

Iris flowers

Edgar was studying Iris flowers, and measured four size variables on each of 50 flowers from each species

What are the main sources of variation across the three species of Iris?

Are there any key characteristics of flowers that differentiate the three species?

One way to make a start on this is to visualise the data via ordination.

Using the princomp function

```
> data("iris")
> pc = princomp(iris[,1:4], cor=TRUE)
> pc
Call:
princomp(x = iris[, 1:4], cor = TRUE)
```

Standard deviations:

Comp.1	Comp.2	Comp.3	Comp.4
1.7083611	0.9560494	0.3830886	0.1439265

4 variables and 150 observations.

Variances (square of standard deviation) of the principal components can be understood as partitioning the total variance in the data (total variance equals 4, the number of response variables). So most of the variation in the data is explained by the first two principal components ($1.7^2/4 = 73\%$ and $0.96^2/4 = 23\%$).

Principal component loadings

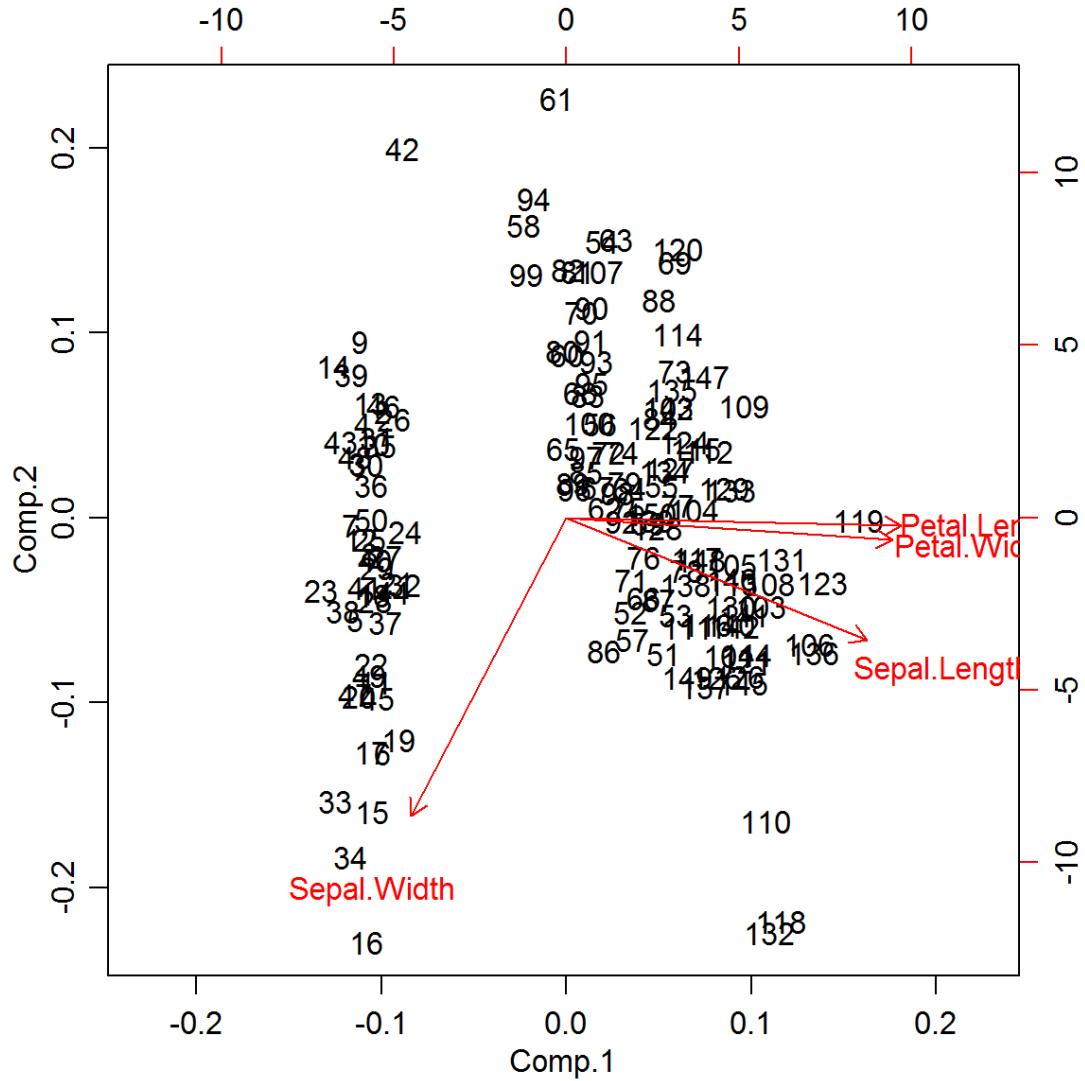
```
> loadings(pc)
```

Loadings:

	Comp.1	Comp.2	Comp.3	Comp.4
Sepal.Length	0.521	-0.377	0.720	0.261
Sepal.Width	-0.269	-0.923	-0.244	-0.124
Petal.Length	0.580		-0.142	-0.801
Petal.Width	0.565		-0.634	0.524

```
> biplot(pc)
```

Any loading less than 0.1 is not shown. The first PC is mostly about petals – higher values for bigger petals, in width and length. The second PC is mostly about sepal width.

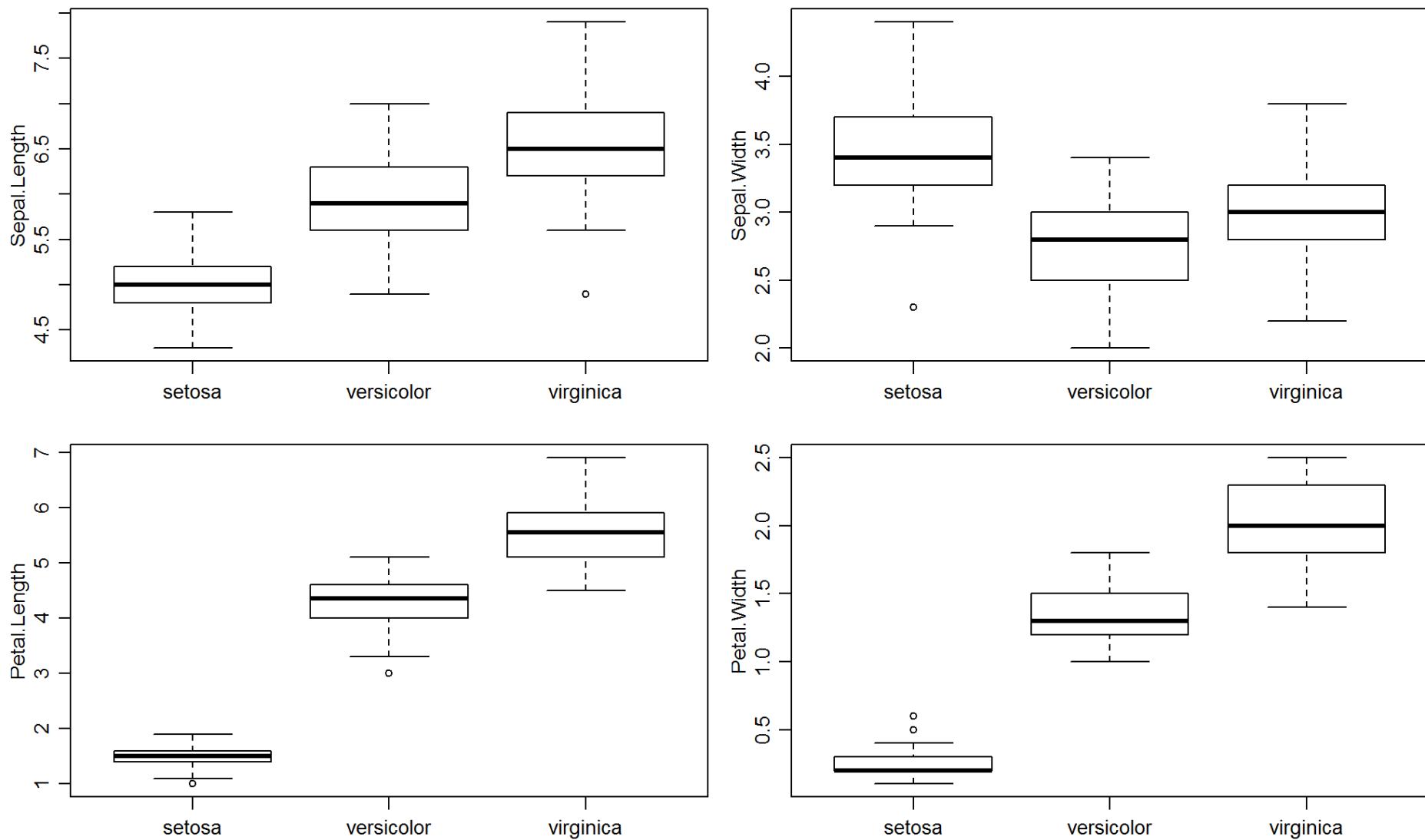


Can you see the three species? Which variables seem most associated with cross-species differences?

But you can also see this from looking at descriptive statistics for each marginal variable:

```
> by(iris, iris$Species, function(dat){ apply(dat[,1:4], 2, mean) } )
iris$Species: setosa
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
      5.006        3.428       1.462        0.246
-----
iris$Species: versicolor
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
      5.936        2.770       4.260        1.326
-----
iris$Species: virginica
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
      6.588        2.974       5.552        2.026

> par(mfrow=c(2,2))
> plot(Sepal.Length~Species, data=iris, xlab="")
> plot(Sepal.Width~Species, data=iris, xlab="")
> plot(Petal.Length~Species, data=iris, xlab="")
> plot(Petal.Width~Species, data=iris, xlab="")
```



Multi-dimensional scaling

(non-metric) multi-dimensional scaling (“MDS” or “nMDS”)

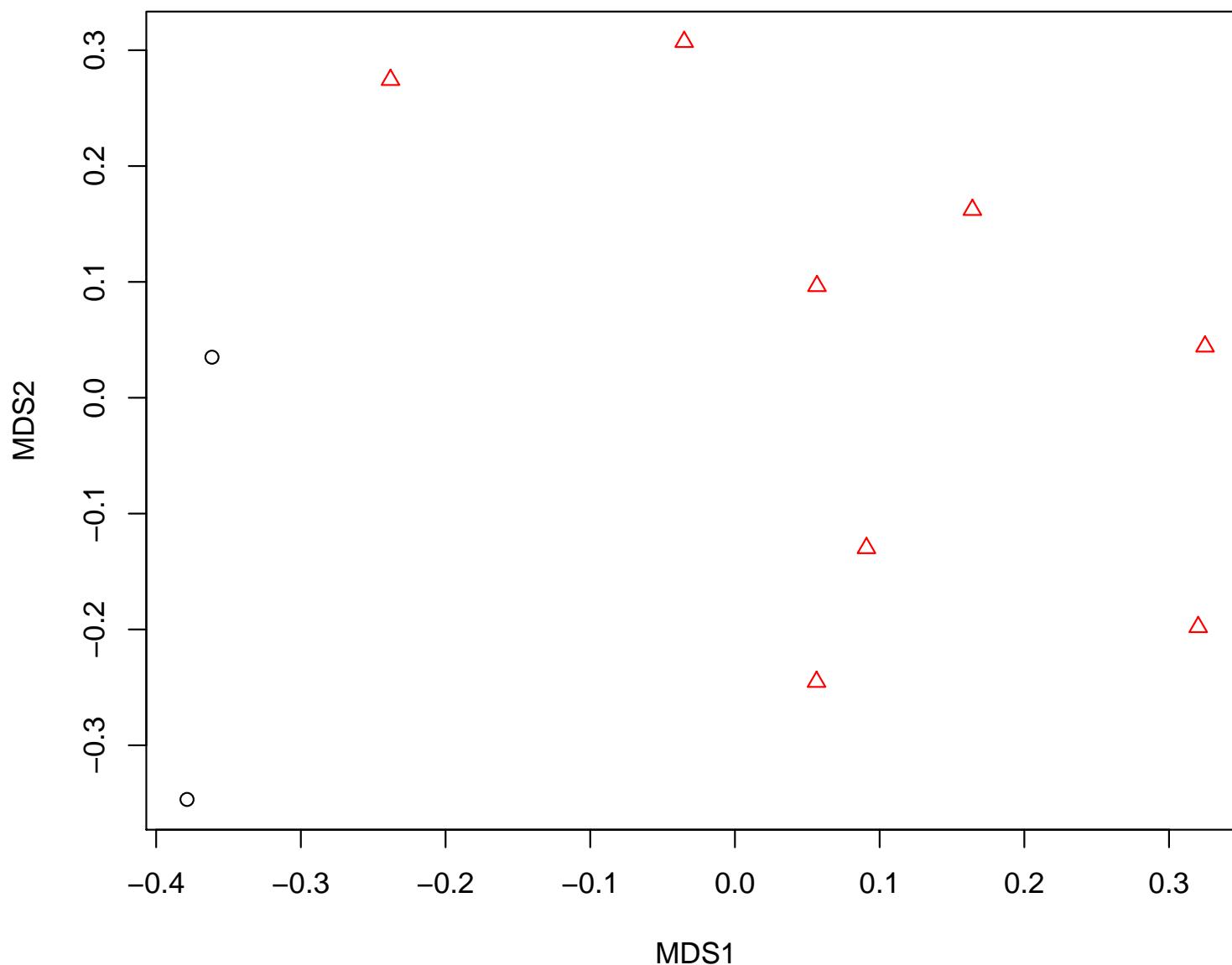
Goal: to capture the “main characteristics” of the data in a 2D plot

- Transform your data (if required/desired)
- Choose some pair-wise dissimilarity measure
- Find some arrangement of points such that pairwise distances on the plot match pair-wise dissimilarities as closely as possible.

```
> library(vegan)
> ord.mds=metaMDS(reveg)
Square root transformation
Wisconsin double standardization
Run 0 stress 0.1611245
Run 1 stress 0.2081913
Run 2 stress 0.1611238
... New best solution
... procrustes: rmse 0.0005853065 max resid 0.001039715
*** Solution reached
```

```
> plot(ord.mds$points,pch=as.numeric(treatment),col=treatment)
```

By default, the vegan package uses Bray-Curtis distances, and a bunch of fairly arbitrary transformations.



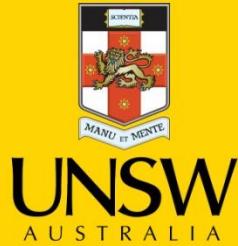
MDS issues

- Transform your data – **how?**
- Choose some pair-wise dissimilarity measure – **how?**

And **how can you check** that you did the right thing?

And **what does this really tell us about correlation** between response variables anyway?

See Session 11 Exercise 1 for a **cautionary tale** about how standard practice for MDS can go really wrong.



The case of the missing model: The modernisation of multivariate analysis in ecology

David Warton

Never Stand Still

Science

School of Mathematics and Statistics



The case of the missing model: The modernisation of multivariate analysis in ecology

- Introduction
- The Bray-Curtis distance and other 1980's memorabilia
- Building a model for multivariate abundances in ecology
- A 2020 vision for multivariate analysis in ecology

An exciting time to do statistics!



<http://www.allaboutapple.com>

Speed: 1MHz RAM: 64kB

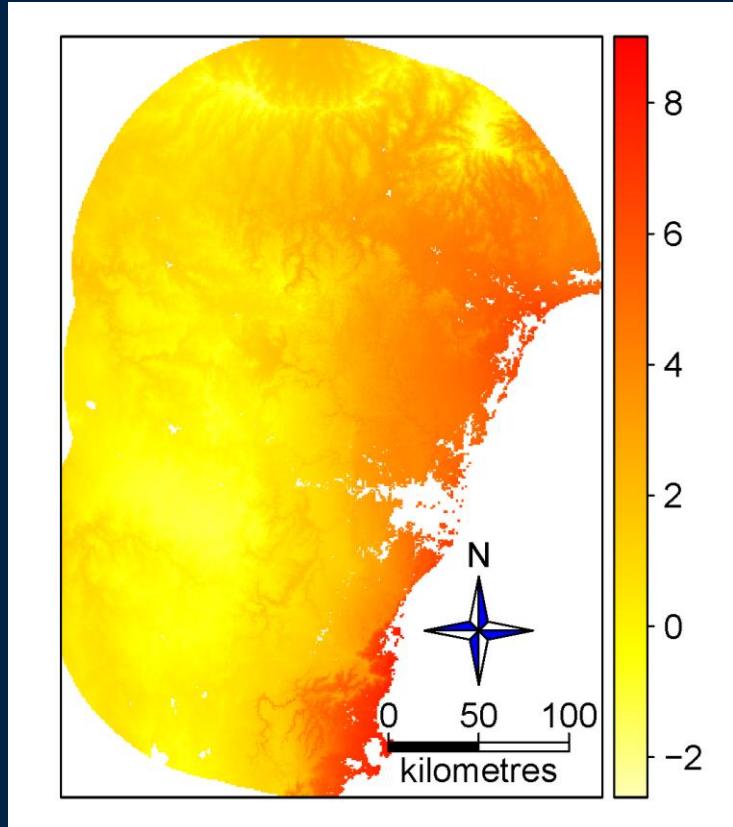


Matthew Riegler, CC-BY, Wikimedia Commons

Speed: 3.8GHz RAM: 16GB

Computing revolution → Data analysis revolution

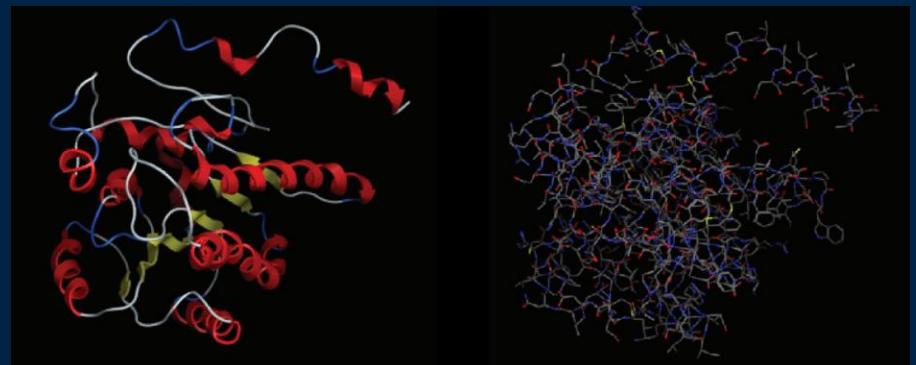
New technology → New data



Warton & Shepherd (2010) *Annals of Applied Statistics*

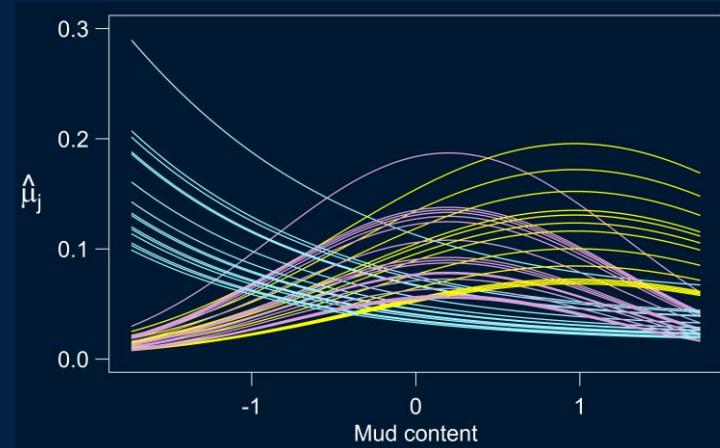
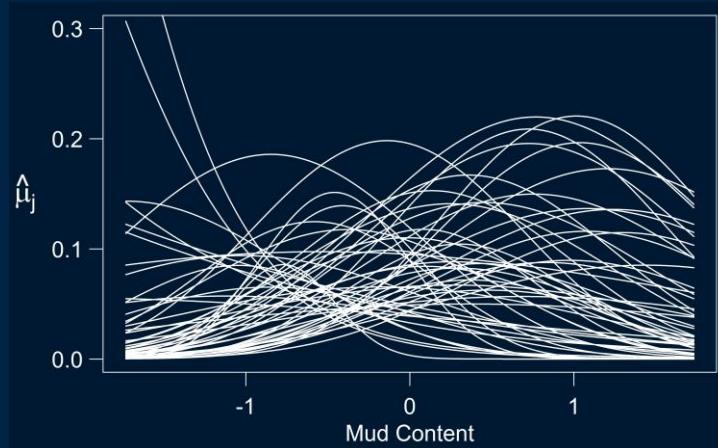
OsAP2/EREBP-041	HLNFP	DLAHLLPRAAS-A
OsAP2/EREBP-007	HLNFP	DLAGVLPLPRAAS-A
OsAP2/EREBP-082	VLNFP	DMAASLPLRPAS-A
OsAP2/EREBP-056	VLNFP	GAAASLPLRPAS-A
OsAP2/EREBP-002	VLNFP	DLAPALPLRPAS-L
OsAP2/EREBP-126	VLNFP	GATASRVPVPAS-A
OsAP2/EREBP-115	ALNFP	DAARSRPAPAS-A
OsAP2/EREBP-136	ALNFP	GTATSRPAPAS-G
OsAP2/EREBP-139	DLNFP	VHLPFHIP-AAA-M
OsAP2/EREBP-138	-LNFA	DSPRRLRVPPIGA
OsAP2/EREBP-081	ELNFP	DSPSTLPRART-A
OsAP2/EREBP-157	RLNFP	AIAHRFRRPDT-A

Rashid et al. (2012) *Evolutionary Bioinformatics*

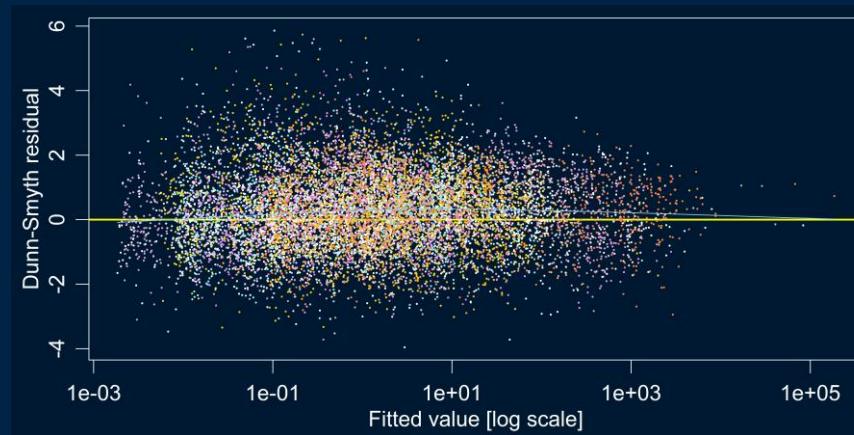


Yamaguchi et al. (2011) *Cancer Informatics*

New technology → New analysis methods

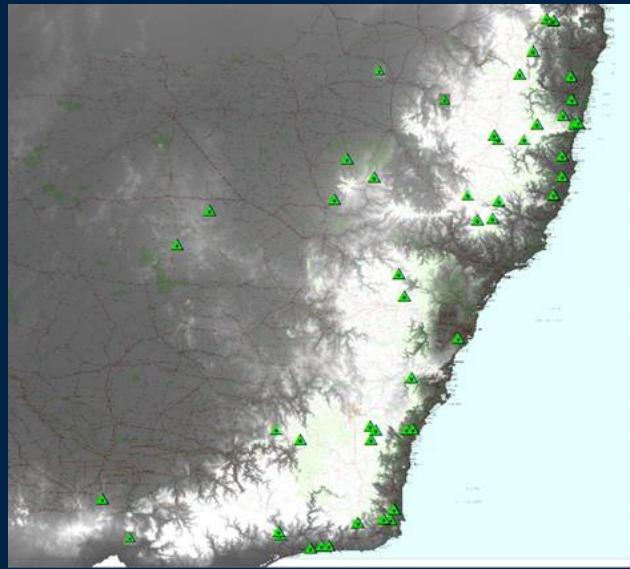


Francis Hui (2014)



Dunstan et al. (2013) *Journal of Agricultural, Biological and Environmental Statistics*

Example: multivariate abundances in ecology



Widely used to study communities.

How do grassland invertebrates respond to climate?

← 64 *Hemiptera* species →

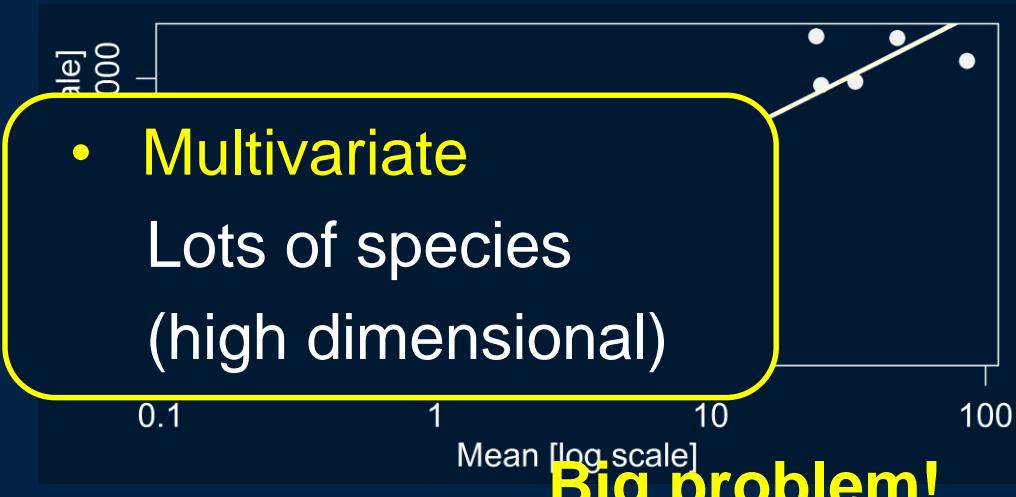
↑ 36 sites ↓

0	5	0	1	0	0	0	0	0	...	0
0	1	0	0	0	0	0	0	1	...	1
0	2	0	0	0	0	0	0	0	...	0
2	0	0	0	1	0	0	0	0	...	0
0	0	0	0	0	0	0	0	0	...	0
0	0	0	0	0	0	0	0	0	...	0
3	0	0	0	0	0	0	0	0	...	2
0	0	0	0	0	0	0	0	3	0	...
0	0	0	0	0	0	0	0	0	0	...
4	0	0	0	0	0	0	0	0	0	...
0	0	0	0	0	0	0	0	0	0	...
0	0	0	0	0	0	1	0	0	...	0
0	0	0	0	0	2	0	0	0	...	0
0	3	1	0	0	6	1	0	0	...	0
0	0	1	1	12	1	0	0	0	...	0
...
4	1	0	0	0	1	0	0	0	...	0

Multivariate abundances: two key properties

- Abundance

Different species, different variability
(mean-variance relationship)



0	5	0	1	0	0	0	0	0	0	...	0
0	1	0	0	0	0	0	0	0	1	...	1
0	2	0	0	0	0	0	0	0	0	...	0
2	0	0	0	1	0	0	0	0	0	...	0
0	0	0	0	0	0	0	0	0	0	...	0
0	0	0	0	0	0	0	0	0	0	...	0
0	0	0	0	0	0	0	0	0	0	...	0
3	0	0	0	0	0	0	0	0	0	...	2
0	0	0	0	0	0	0	0	3	0	...	0
0	0	0	0	0	0	0	0	0	0	...	0
4	0	0	0	0	0	0	0	0	0	...	0
0	0	0	0	0	0	0	0	0	0	...	0
0	0	0	0	0	0	0	1	0	0	...	0
0	0	0	0	0	0	2	0	0	0	...	0
0	3	1	0	0	6	1	0	0	0	...	0
0	0	1	1	12	1	0	0	0	0	...	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
4	1	0	0	0	1	0	0	0	0	...	0

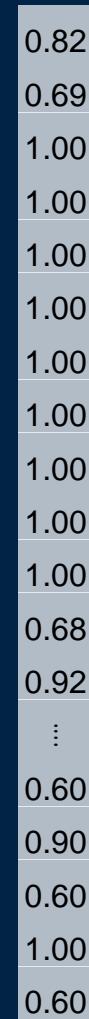
The case of the missing model: The modernisation of multivariate analysis in ecology

- Introduction
- The Bray-Curtis distance and other 1980's memorabilia
- Building a model for multivariate abundances in ecology
- A 2020 vision for multivariate analysis in ecology

1980's solution...

0	5	0	1	0	0	0	0	0	...	0
0	1	0	0	0	0	0	0	1	...	1
0	2	0	0	0	0	0	0	0	...	0
2	0	0	0	1	0	0	0	0	...	0
0	0	0	0	0	0	0	0	0	...	0
0	0	0	0	0	0	0	0	0	...	0
0	0	0	0	0	0	0	0	0	...	0
3	0	0	0	0	0	0	0	0	...	2
0	0	0	0	0	0	0	3	0	...	0
0	0	0	0	0	0	0	0	0	...	0
4	0	0	0	0	0	0	0	0	...	0
0	0	0	0	0	0	0	0	0	...	0
0	0	0	0	0	0	1	0	0	...	0
0	0	0	0	0	2	0	0	0	...	0
0	3	1	0	0	6	1	0	0	...	0
0	0	1	1	12	1	0	0	0	...	0
...
4	1	0	0	0	1	0	0	0	...	0

$$d_{ii'} = \frac{\sum_j |y_{ij} - y_{i'j}|}{\sum_j (y_{ij} + y_{i'j})}$$



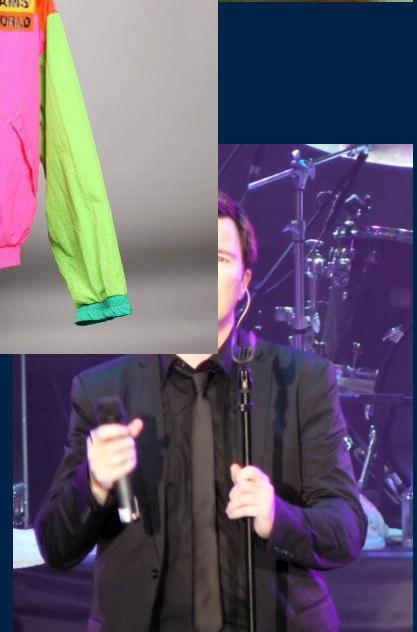
Widely used – often taught in second year ecology!

Some other 1980's trends...

0	5	0	1	0	0	0	0	0	0	...	0
0	1	0	0	0	0	0	0	1	...	1	
0	2	0	0	0	0	0	0	0	...	0	
2	0	0	0	1	0	0	0	0	...	0	
0	0	0	0	0	0	0	0	0	...	0	
0	0	0	0	0	0	0	0	0	...	0	
3	0	0	0	0	0	0	0	0	...	2	
0	0	0	0	0	0	0	3	0	...	0	
0	0	0	0	0	0	0	0	0	...	0	
4	0	0	0	0	0	0	0	0	...	0	
0	0	0	0	0	0	0	0	0	...	0	
0	0	0	0	0	0	1	0	0	...	0	
0	0	0	0	0	2	0	0	0	...	0	
0	3	1	0	0	6	1	0	0	...	0	
0	0	1	1	1	12	1	0	0	...	0	
...	
4	1	0	0	0	0	1	0	0	...	0	



$$d_{ii'} = \frac{\sum_j |y_{ij} - y_{i'j}|}{\sum_j (y_{ij} + y_{i'j})}$$



1980's solution... no good in the 2010's

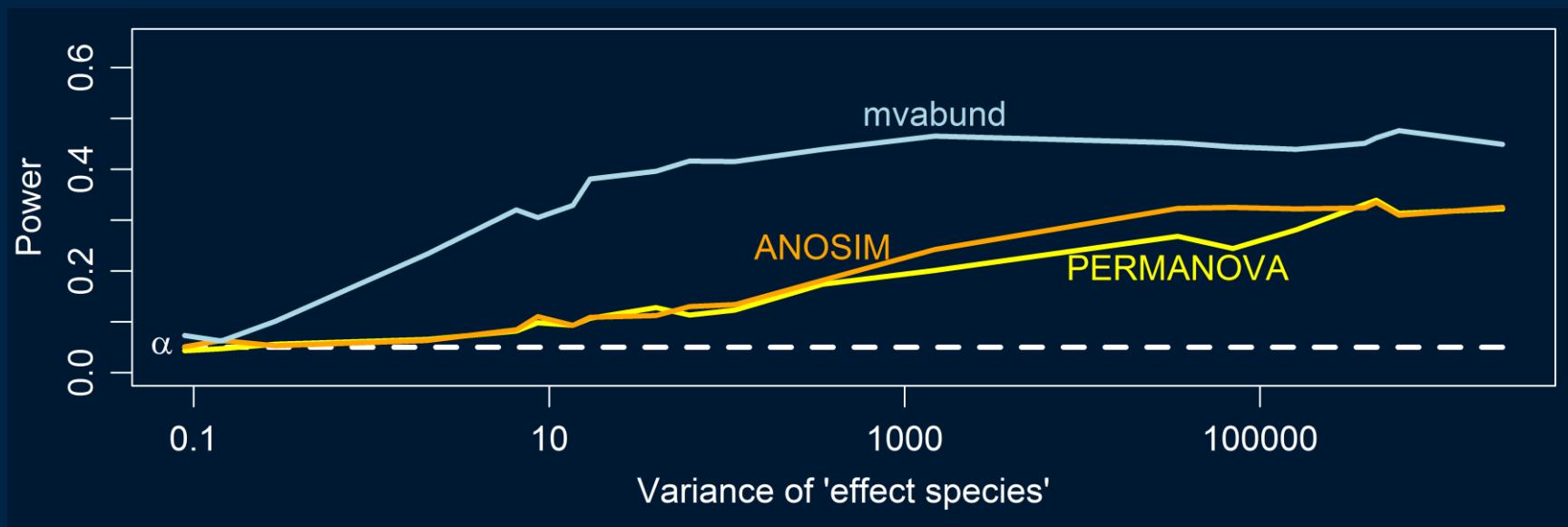
Poor power properties (*transformation doesn't help rarer spp*)
(because mean-variance relationship ignored)



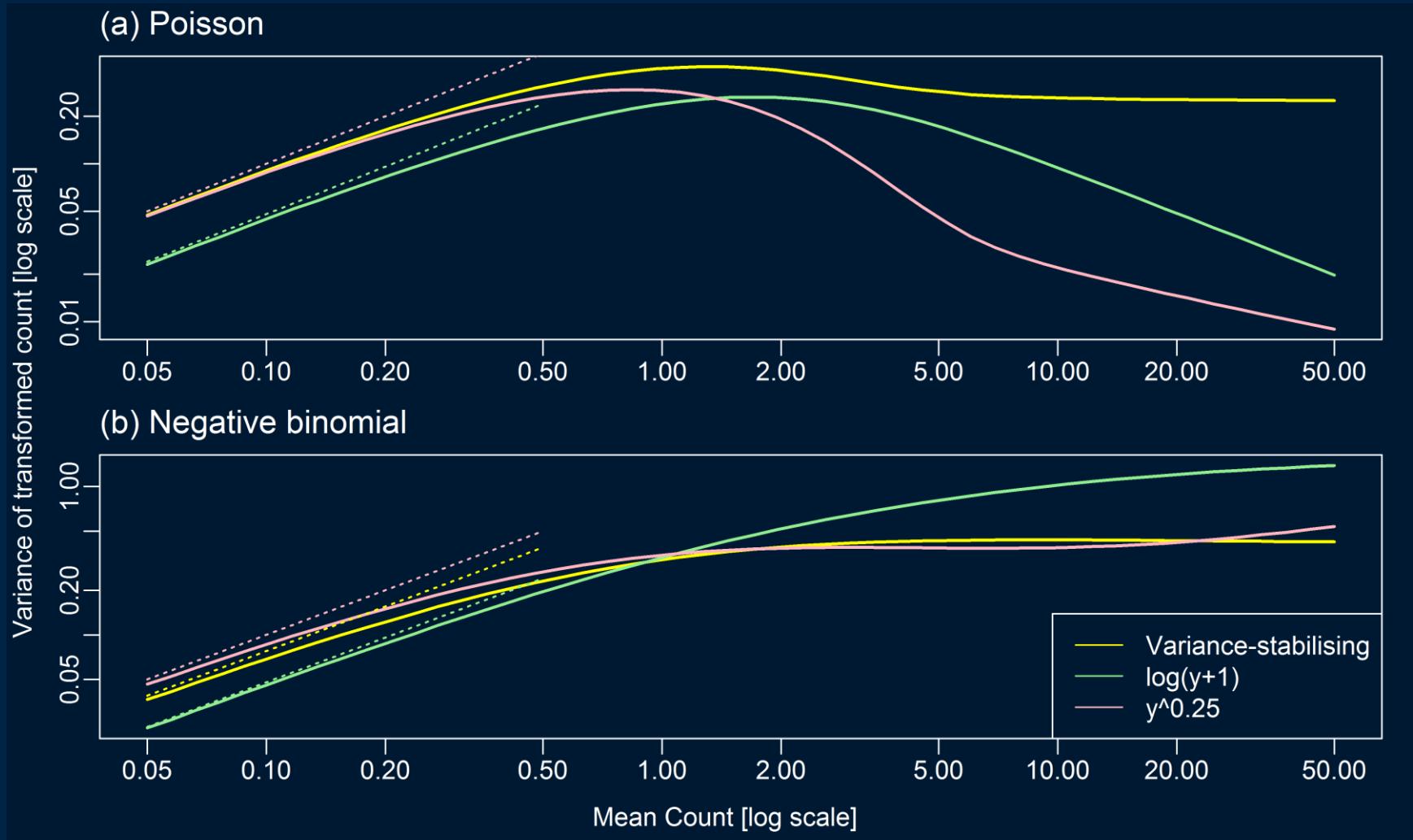
Simulation: a large (10x) effect in one species only.

Results: effect only detected in high-variance species.

Warton et al. (2012) *Methods in Ecology & Evolution*

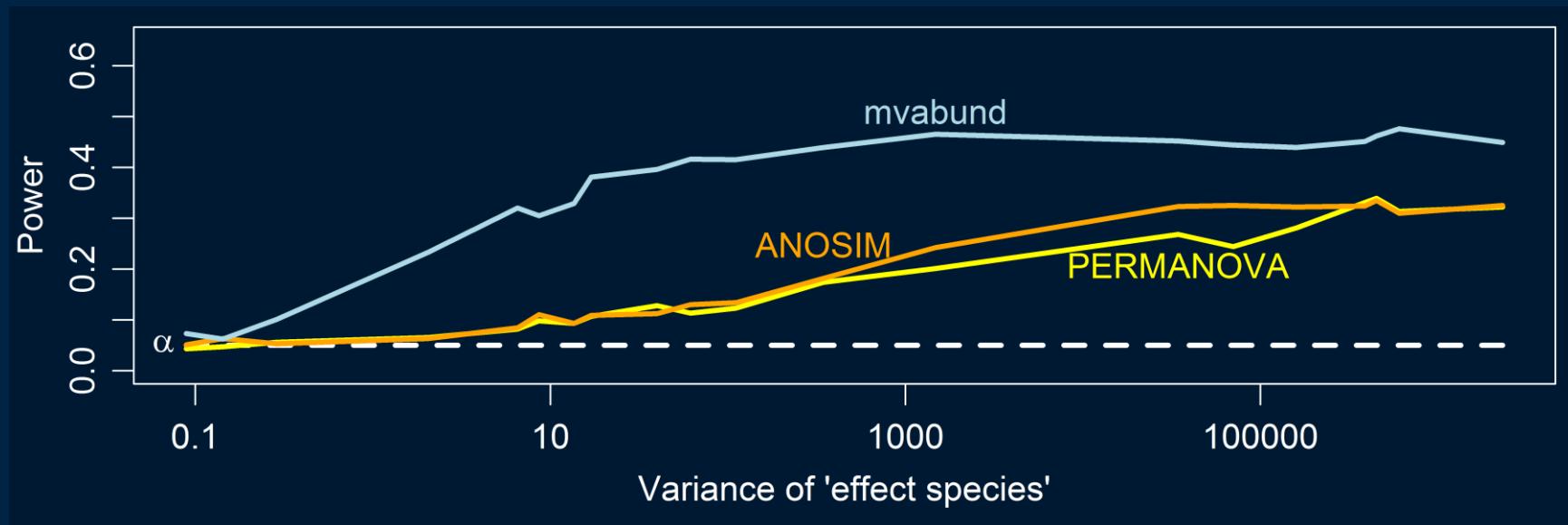


Transformation never works for rare counts



1980's solution... no good in the 2010's

You can't beat poor power properties by transformation
(if you have lots of zeros)



1980's solution... no good in the 2010's

Can't detect special mean structures
(because no mean model)

e.g. evidence of a change in *composition*?

$$H_0 : \boldsymbol{\mu}_i = \alpha_i \boldsymbol{\mu} \quad \text{where } \boldsymbol{\mu}_i = (\mu_{i1}, \dots, \mu_{ip})$$

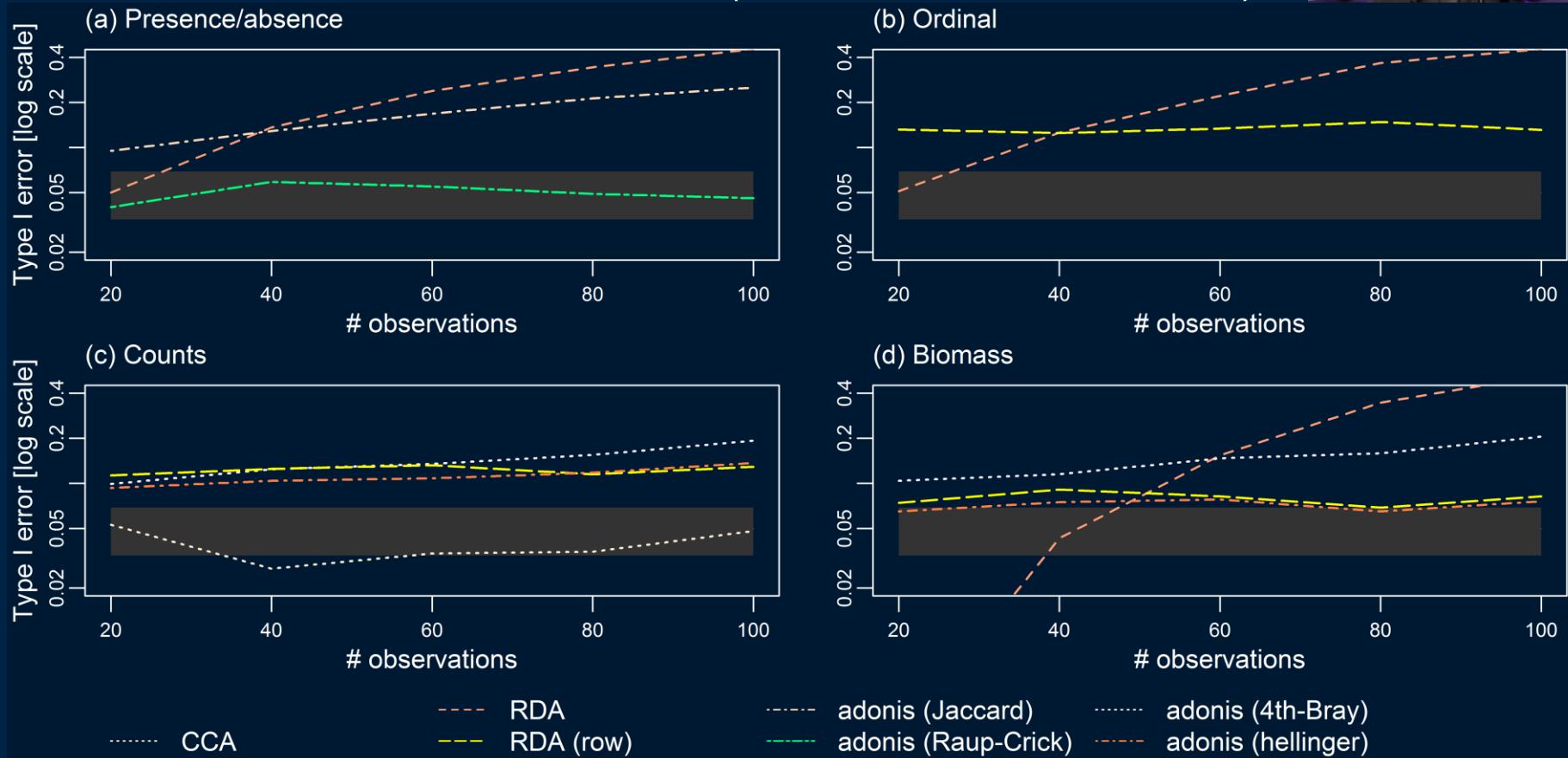
Simulation: 2-fold change in total abundance, no change in composition

Warton et al. (in prep)

1980's solution... no good in the 2010's

Can't detect special mean structures

(because no mean model)



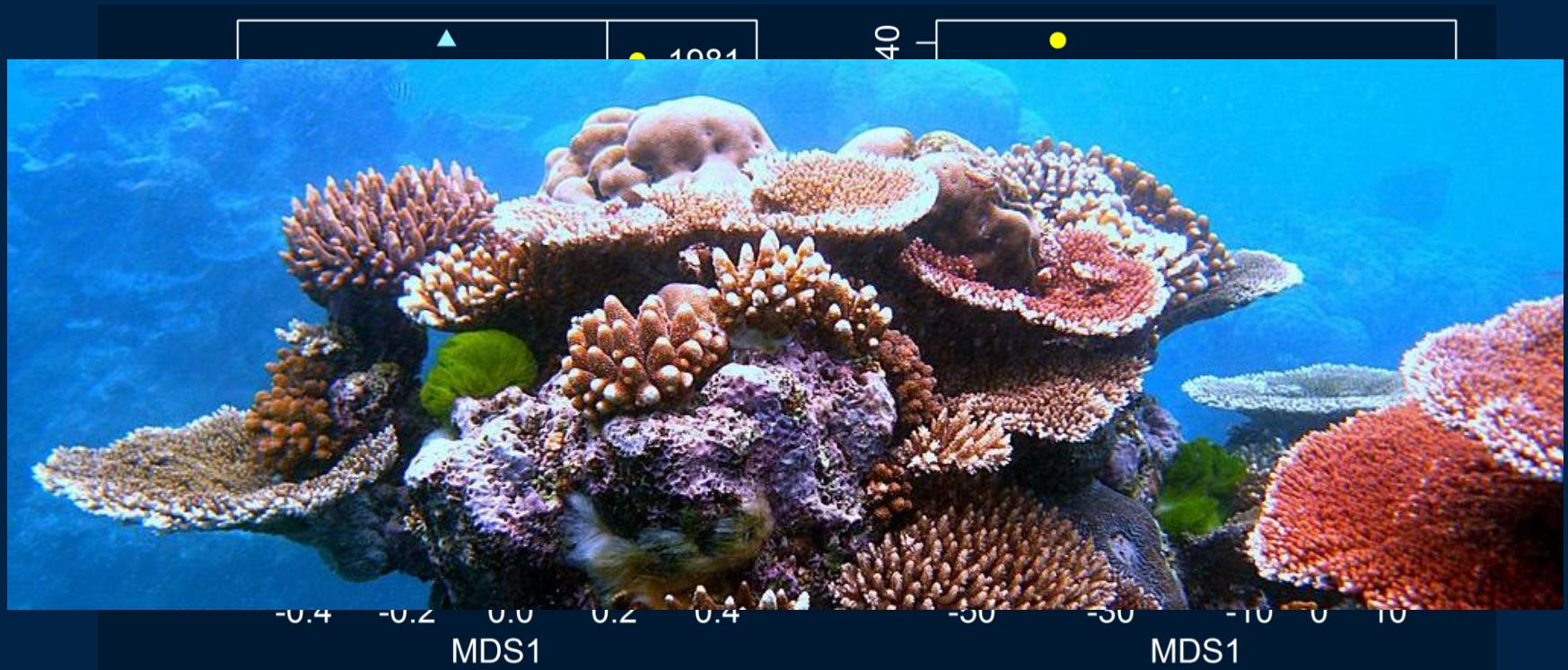
1980's solution... no good in the 2010's

Same dataset, same question, different answers?

(lacks diagnostic tools)

e.g. coral abundance in the Thousand Islands, Indonesia

(Warwick *et al.* 1990)



What's missing from this picture?



for the data-generating mechanism

The case of the missing model: The modernisation of multivariate analysis in ecology

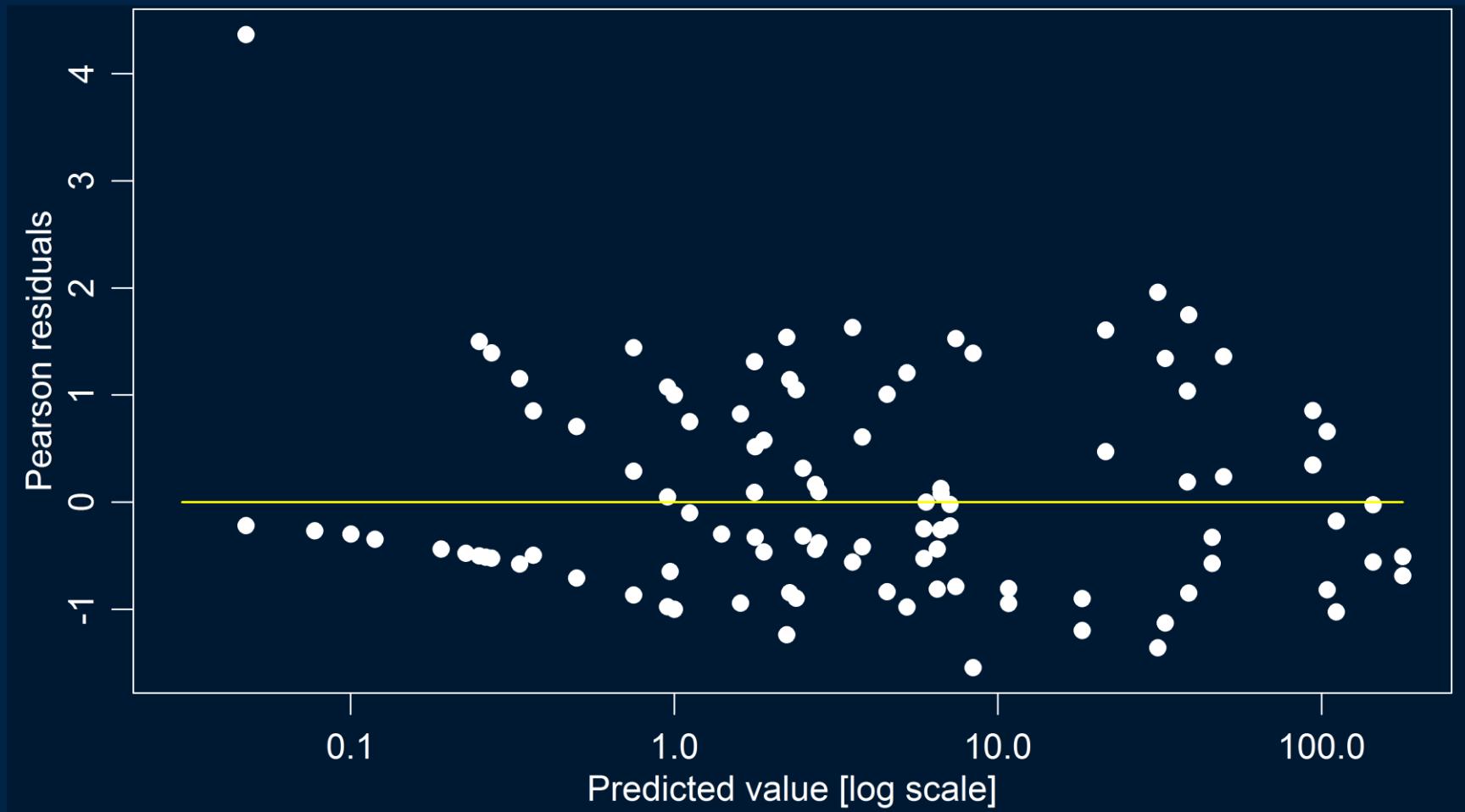
- Introduction
- The Bray-Curtis distance and other 1980's memorabilia
- Building a model for multivariate abundances in ecology
- A 2020 vision for multivariate analysis in ecology

Some things that happened since the 1980's...

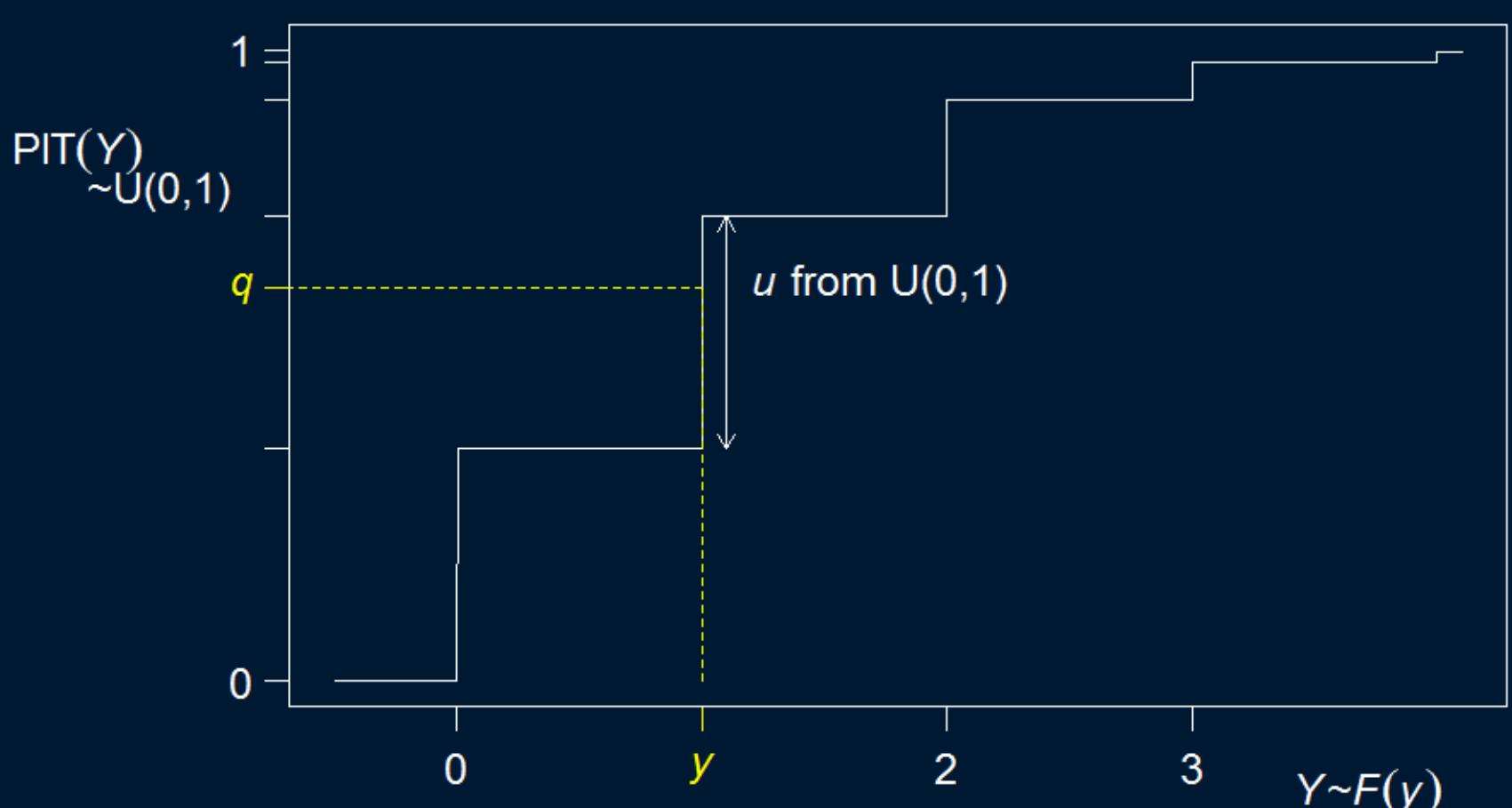
- Bayesian hierarchical models (MCMC) (Gelfand and Smith 1990)
- Monte Carlo EM (Wei & Tanner 1990) and modern numerical integration
- PIT residuals as diagnostics for GLMs (Dunn & Smyth 1996)



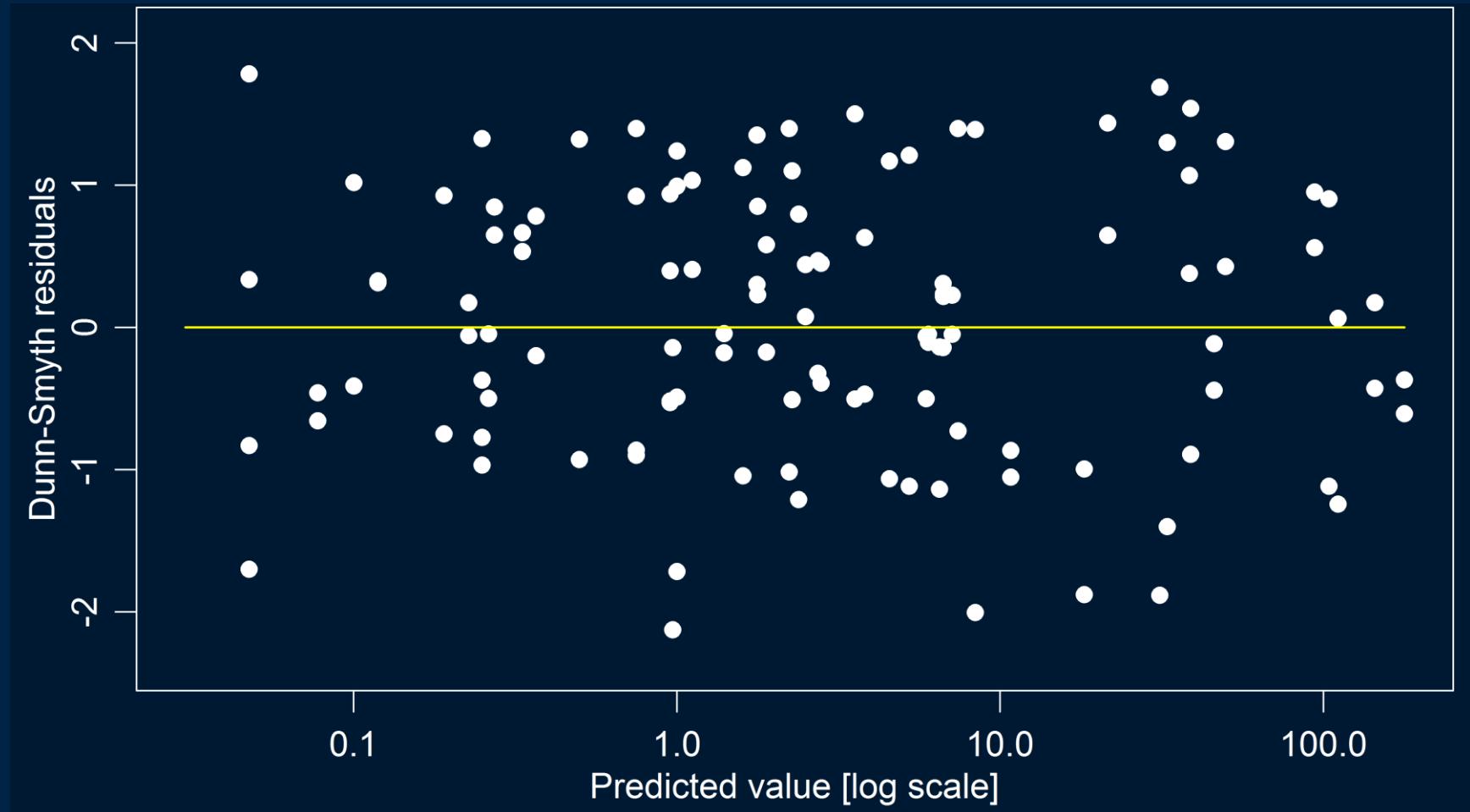
Residuals are hard to define for count data



PIT residuals $\sim U(0,1)$ when model is correct



We can use PIT residuals to check model fit



Some things that happened since the 1980's...

- Bayesian hierarchical models (MCMC) (Gelfand and Smith 1990)
- Modern numerical integration, Monte Carlo EM (Wei & Tanner 1990)
- PIT residuals to check model fit for counts (Dunn & Smyth 1996)
- Penalised likelihood (e.g. LASSO, Tibshirani 1996)
- Generalised linear latent variable models (Skrondal & Rabe-Hesketh 2004)
- Graphical modelling (Friedman *et al.* 2008)
- Variational approximation (Ormerod & Wand 2010)



Things to consider when building a model

- What is the research **Question**?
 - descriptive, prediction, testing *a priori* hypothesis, ...
- What **Properties** do my data have?
 - abundance (mean-variance)
 - multivariate (high-dimensional)



The question?



Data properties?

Abundance
(mean-variance)

Multivariate
(high-dimensional)

Test (env impact)

Composition

Explain spp
variation

Predict
(new sites or spp)

Classify

Ordination

Study correlations

The question?



- Test (env impact)
- Composition
- Explain spp variation
- Predict (new sites or spp)
- Classify
- Ordination
- Study correlations

Data properties?

Abundance
(mean-variance)

Multivariate
(high-dimensional)

Generalised linear model (GLM)

$$\text{Var}(y_{ij}) = V(\mu_{ij}, \phi_j)$$

$$g(\mu_{ij}) = \mathbf{x}'_i \boldsymbol{\beta}_j$$

for abundance y_{ij} of species j at site i
as a function of site predictors \mathbf{x}_i

The question?



- Test (env impact)
- Composition
- Explain spp variation
- Predict (new sites or spp)
- Classify
- Ordination
- Study correlations

Data properties?

Abundance
(mean-variance)

Multivariate
(high-dimensional)

Multivariate frameworks:

- Generalised estimating equations (GEE)
(Warton 2011 *Biometrics*)
- Hierarchical models
(Warton et al. 2015 *Trends Ecol Evol*)
$$\mathbf{u}(\beta) = \sum_{i=1}^n \mathbf{D}_i V(\mu_i, \mathbf{R})^{-1} (\mathbf{y}_i^{(Warton et al. 2015 Trends Ecol Evol)} - \mu_i)$$
 where $\mathbf{D}_i = \frac{\partial \mu_i}{\partial \beta}$
- Copulas
(Popovic & Warton in review)
$$\text{Var}(y_{ij}|m_{ij}) = V(m_{ij}, \phi_j)$$

But high-dimensionality is the real issue
(64 species, 2016 correlations)

The question?



- Test (env impact)
- Composition
- Explain spp variation
- Predict (new sites or spp)
- Classify
- Ordination
- Study correlations

Data properties?

Abundance
(mean-variance)

Multivariate
(high-dimensional)

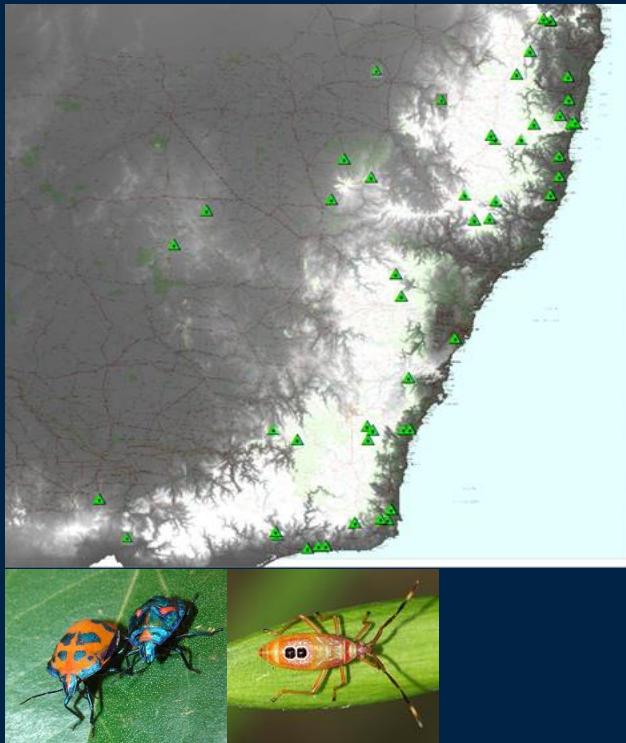
Options for high dimensionality:

- Ignore it (e.g. for prediction)
Brown et al (2014) *Methods Ecol Evol*
- Resample rows for valid inference
Wang et al (2012) *Methods Ecol Evol*
- Shrink covariance (e.g. toward identity)
Warton (2008) *JASA*
- Covariance modelling, e.g.
 - Factor analysis (latent variables)
Warton et al (2015) *Trends Ecol Evol*
 - Graphical modelling

The question?



Test (env impact)



Data properties?

Abundance
(mean-variance)

Multivariate
(high-dimensional)

Do grassland invertebrate communities change when host plants are transplanted?

64 <i>Hemiptera</i> species												
36 sites												
0	5	0	1	0	0	0	0	0	0	...	0	0
0	1	0	0	0	0	0	0	0	1	...	1	0
0	2	0	0	0	0	0	0	0	0	...	0	0
2	0	0	0	1	0	0	0	0	0	...	0	0
0	0	0	0	0	0	0	0	0	0	...	0	0
0	0	0	0	0	0	0	0	0	0	...	0	0
0	0	0	0	0	0	0	0	0	0	...	0	0
3	0	0	0	0	0	0	0	0	0	...	2	0
0	0	0	0	0	0	0	0	0	3	0	...	0
0	0	0	0	0	0	0	0	0	0	...	0	0
4	0	0	0	0	0	0	0	0	0	...	0	0
0	0	0	0	0	0	0	0	0	0	...	0	0
0	0	0	0	0	0	0	1	0	0	...	0	0
0	0	0	0	0	2	0	0	0	0	...	0	0
0	3	1	0	0	6	1	0	0	0	...	0	0
0	0	1	1	12	1	0	0	0	0	...	0	0
...
4	1	0	0	0	1	0	0	0	0	...	0	0

The question?



Test (env impact)

Composition

Explain spp
variation

Predict
(new sites or spp)

Classify

Ordination

Study correlations

Data properties?

Abundance
(mean-variance)

Multivariate
(high-dimensional)

Generalised linear model (GLM)

$$g(\mu_{ij}) = \mathbf{x}'_i \boldsymbol{\beta}_j$$

$$\text{Var}(y_{ij}) = V(\mu_{ij}, \phi_j)$$

for abundance y_{ij} of species j at site i
as a function of site predictors \mathbf{x}_i

The question?



Test (env impact)

Composition

Explain spp
variation

Predict
(new sites or spp)

Classify

Ordination

Study correlations

Data properties?

Abundance
(mean-variance)

Multivariate
(high-dimensional)

Resample rows for testing

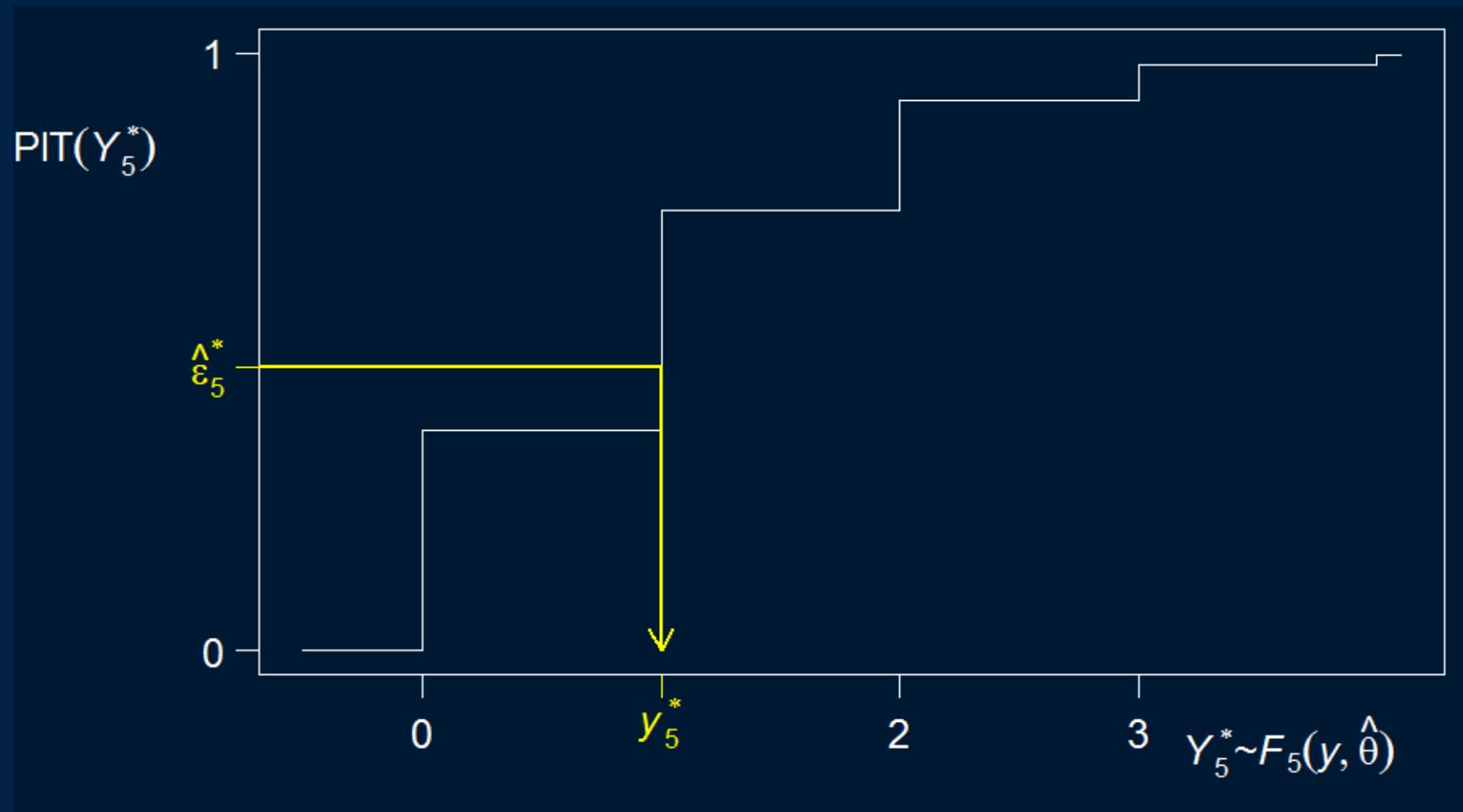
How to resample GLMs?

“PIT-trap” for residual resampling

(Warton & Wang in review)

PIT-trap: Solve $F(y_i^*, \hat{\theta}) = \hat{\epsilon}_i^*$ for y_i^*

Warton & Wang (in review)



The question?



Test (env impact)

Composition

Explain spp
variation

Predict
(new sites or spp)

Classify

Ordination

Study correlations

Data properties?

Abundance
(mean-variance)

Multivariate
(high-dimensional)

row-resample GLMs

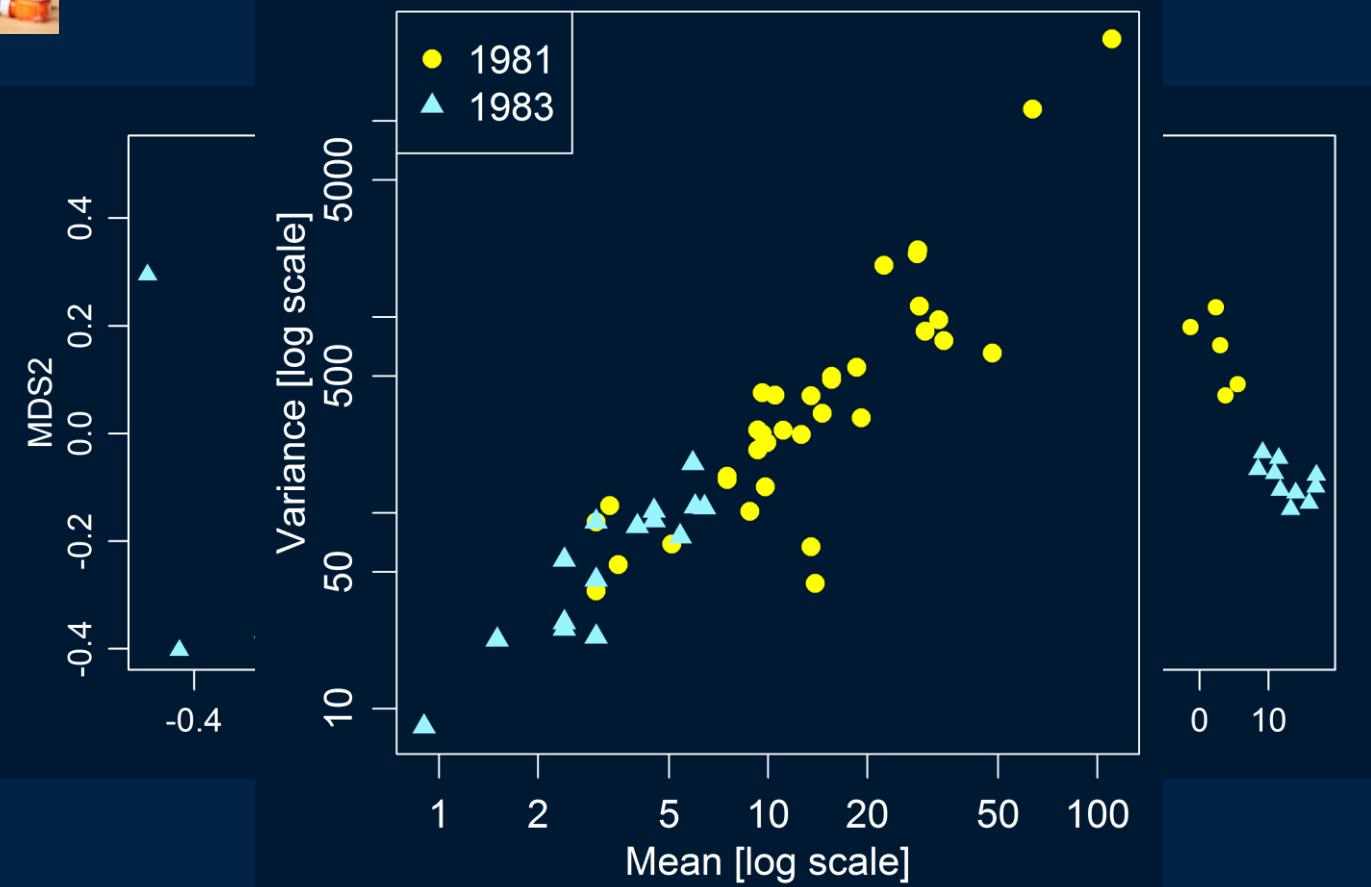
The question?



Ordination

Abundance
(mean-variance)

Multivariate
(high-dimensional)



The question?



Data properties?

Abundance
(mean-variance)

Multivariate
(high-dimensional)

Generalised linear model (GLM)

$$g(\mu_{ij}) = \mathbf{x}'_i \boldsymbol{\beta}_j$$

$$\text{Var}(y_{ij}) = V(\mu_{ij}, \phi_j)$$

Ordination

The question?



Data properties?

Abundance
(mean-variance)

Multivariate
(high-dimensional)

Generalised linear model (GLM)
with latent variables z_i

$$g(\mu_{ij}) = \mathbf{x}'_i \boldsymbol{\beta}_j + \mathbf{z}'_i \boldsymbol{\lambda}_j$$

$$\text{Var}(y_{ij}) = V(\mu_{ij}, \phi_j)$$

$$\mathbf{z}_{ij} \stackrel{\text{iid}}{\sim} N(0, 1)$$

Warton et al. (*Trends Ecol Evol*)

Ordination

The computation of ordination axes

- MCMC (Hui & Warton, Chiaromonte & Blanchet (MS))
- variational approximation

Hui et al (in press) *J Comp Graph Stat*

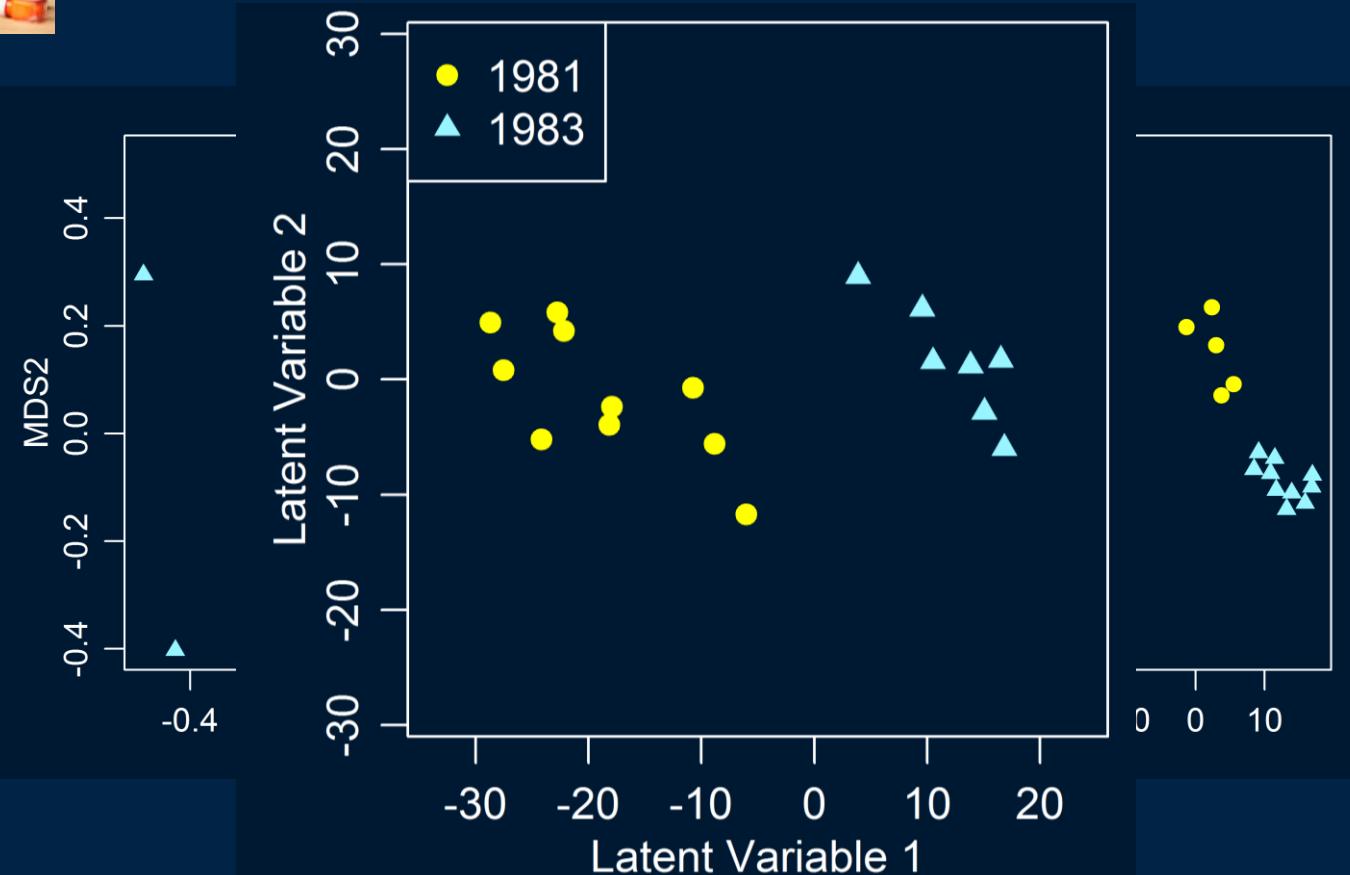
The question?



Ordination

Abundance
(mean-variance)

Multivariate
(high-dimensional)



Hui et al. (2015) *Methods Ecol Evol*

The question?



Data properties?

Abundance
(mean-variance)

Multivariate
(high-dimensional)

Test (env impact) row-resample GLMs

Composition

Explain spp
variation

Predict
(new sites or spp)

Classify

Ordination

Study correlations

Latent variable models

also see Pledger & Arnold (2014, *J Comp Graph Stats*)

The question?



- Test (env impact)
- Composition
- Explain spp variation
- Predict (new sites or spp)
- Classify
- Ordination
- Study correlations

Abundance
(mean-variance)

Multivariate
(high-dimensional)

Is there a change in community composition? (relative abundance)

	64 <i>Hemiptera</i> species										
↑	0	5	0	1	0	0	0	0	0	...	0
36 sites ↓	0	1	0	0	0	0	0	0	1	...	1
0	2	0	0	0	0	0	0	0	0	...	0
2	0	0	0	1	0	0	0	0	0	...	0
0	0	0	0	0	0	0	0	0	0	...	0
0	0	0	0	0	0	0	0	0	0	...	0
3	0	0	0	0	0	0	0	0	0	...	2
0	0	0	0	0	0	0	0	0	0	...	0
0	0	0	0	0	0	0	0	0	0	...	0
4	0	0	0	0	0	0	0	0	0	...	0
0	0	0	0	0	0	0	0	0	0	...	0
0	0	0	0	0	0	1	0	0	0	...	0
0	0	0	0	0	2	0	0	0	0	...	0
0	3	1	0	0	6	1	0	0	0	...	0
0	0	1	1	12	1	0	0	0	0	...	0
...
4	1	0	0	0	1	0	0	0	0	...	0



Data properties?

Abundance (mean-variance)

Multivariate (high-dimensional)

Test (env impact)

Composition

Explain spp variation

Predict “r” (new sites or spp)

Classify

Ordination

Study correlations

Is there a change in community composition? (relative abundance)

← 64 *Hemiptera* species →

0	5	0	1	0	0	0	0	0	...	0	
0	1	0	0	0	0	0	0	1	...	1	
0	2	0	0	0	0	0	0	0	...	0	
2	0	0	0	1	0	0	0	0	...	0	
0	0	0	0	0	0	0	0	0	...	0	
0	0	0	0	0	0	0	0	0	...	0	
3	0	0	0	0	0	0	0	0	...	2	
0	0	0	0	0	0	0	0	3	0	...	0
0	0	0	0	0	0	0	0	0	0	...	0
4	0	0	0	0	0	0	0	0	0	...	0
0	0	0	0	0	0	0	0	0	0	...	0
0	0	0	0	0	0	0	1	0	0	...	0
0	0	0	0	0	2	0	0	0	0	...	0
0	3	1	0	0	0	1	0	0	0	...	0
0	0	1	1	12	1	0	0	0	0	...	0
...	
4	1	0	0	0	1	0	0	0	0	...	0

“row effect”,

The question?



Test (env impact)

Composition

Explain spp
variation

Predict
(new sites or spp)

Classify

Ordination

Study correlations

Data properties?

Abundance
(mean-variance)

Multivariate
(high-dimensional)

row-resample GLMs

$$g(\mu_{ij}) = \alpha_i + \mathbf{x}'_i \boldsymbol{\beta}_j$$

Warton *et al.* (in prep)

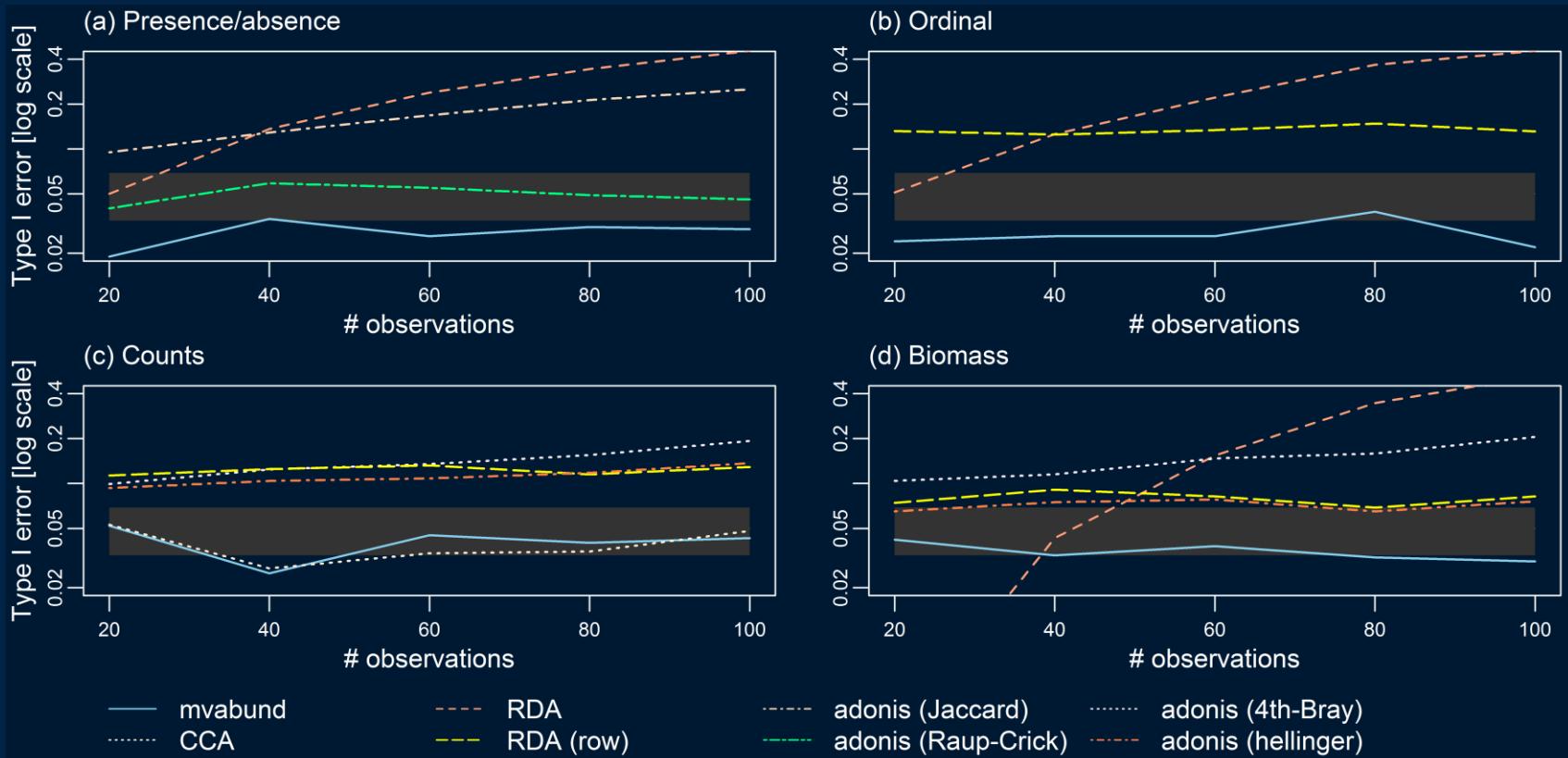
The question?



Data properties?

Abundance
(mean-variance)

Multivariate
(high-dimensional)



The question?



Data properties?

Abundance
(mean-variance)

Multivariate
(high-dimensional)

Composition

Coral data: effect on total abundance or composition?



The question?



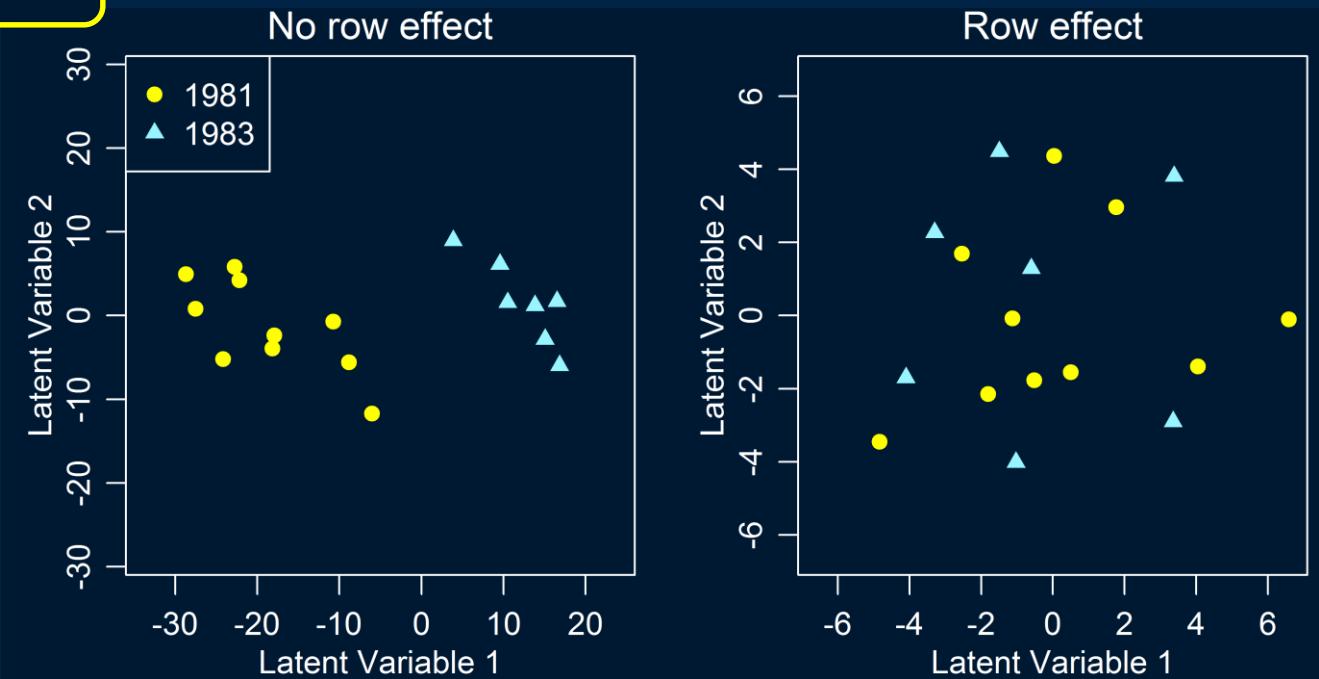
Data properties?

Abundance
(mean-variance)

Multivariate
(high-dimensional)

Composition

Coral data: effect on total abundance or composition?



Data properties?



The question?

- Test (env impact)
- Composition
- Explain spp variation
- Predict (new sites or spp)
- Classify
- Ordination
- Study correlations

Abundance
(mean-variance)

Multivariate
(high-dimensional)

Why do different species respond differently?

64 <i>Hemiptera</i> species													
36 sites													
0	5	0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	...	1	0
0	2	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	2
0	0	0	0	0	0	0	0	0	0	3	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	2	0	0	0	0	0	0	0	0
0	3	1	0	0	6	1	0	0	0	0	0	0	0
0	0	1	1	12	1	0	0	0	0	0	0	0	0
...
4	1	0	0	0	1	0	0	0	0	0	0	0	0

The question?



Test (env impact)

Composition

Explain spp
variation

Predict
(new sites or spp)

Classify

Ordination

Study correlations

Data properties?

Abundance
(mean-variance)

Multivariate
(high-dimensional)

Why do different species respond differently?
→ species traits (\mathbf{z}_j) as predictors

$$g(\mu_{ij}) = \mathbf{x}'_i \boldsymbol{\beta}_x + \mathbf{z}'_j \boldsymbol{\beta}_z + (\mathbf{x}_i \otimes \mathbf{z}_j)' \boldsymbol{\beta}_{xz}$$

climate x trait interactions ($\boldsymbol{\beta}_{xz}$) explain differences
in species response, “the fourth corner”

Brown *et al.* (2014), Warton *et al.* (2015) *Methods in Ecology and Evolution*

The question?

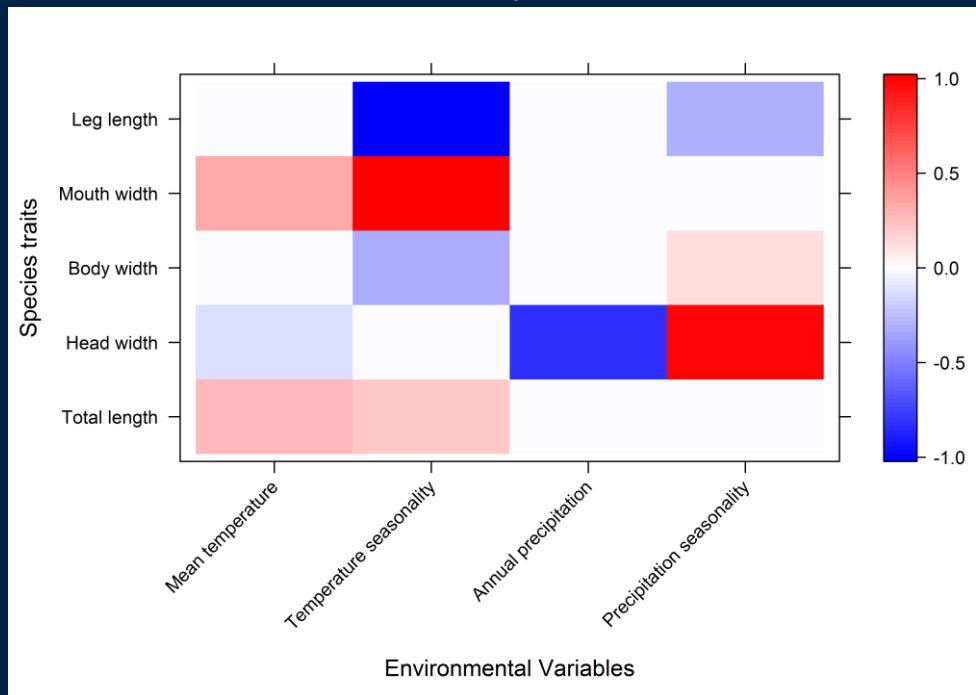
- Test (env impact)
- Composition
- Explain spp variation
- Predict (new sites or spp)
- Classify
- Ordination
- Study correlations



Abundance
(mean-variance)

Multivariate
(high-dimensional)

Why do different species respond differently?
→ species traits (z_j) as predictors



The question?



Test (env impact)

Composition

Explain spp
variation

Predict
(new sites or spp)

Classify

Ordination

Study correlations

Data properties?

Abundance
(mean-variance)

Multivariate
(high-dimensional)

row-resample GLMs

$$g(\mu_{ij}) = \alpha_i + \mathbf{x}'_i \boldsymbol{\beta}_j$$

Traits (\mathbf{z}_i) as predictors, interaction with \mathbf{x}_i :

$$g(\mu_{ij}) = \mathbf{x}'_i \boldsymbol{\beta}_x + \mathbf{z}'_j \boldsymbol{\beta}_z + (\mathbf{x}_i \otimes \mathbf{z}_j)' \boldsymbol{\beta}_{xz}$$

Latent variable models

also see Pledger & Arnold (2014, *J Comp Graph Stats*)

The question?



- Test (env impact)
- Composition
- Explain spp variation
- Predict (new sites or spp) Predict (new sites or spp)
- Classify
- Ordination
- Study correlations

Abundance
(mean-variance)

Multivariate
(high-dimensional)

row-resample GLMs

$$g(\mu_{ij}) = \alpha_i + \mathbf{x}'_i \boldsymbol{\beta}_j$$

Traits (\mathbf{z}_j) as predictors, interaction with \mathbf{x}_i :

$$g(\mu_{ij}) = \mathbf{x}'_i \boldsymbol{\beta}_x + \mathbf{z}'_j \boldsymbol{\beta}_z + (\mathbf{x}_i \otimes \mathbf{z}_j)' \boldsymbol{\beta}_{xz}$$

LASSO, random effects, neural nets, ...

(Harris 2015, *Methods in Ecology and Evolution*)

Latent variable models

also see Pledger & Arnold (2014, *J Comp Graph Stats*)

The question?



- Test (env impact)
- Composition
- Explain spp variation
- Predict (new sites or spp)
- Classify
- Ordination
- Study correlations

Abundance
(mean-variance)

Multivariate
(high-dimensional)

row-resample GLMs

$$g(\mu_{ij}) = \alpha_i + \mathbf{x}'_i \boldsymbol{\beta}_j$$

Traits (\mathbf{z}_j) as predictors, interaction with \mathbf{x}_i :

$$g(\mu_{ij}) = \mathbf{x}'_i \boldsymbol{\beta}_x + \mathbf{z}'_j \boldsymbol{\beta}_z + (\mathbf{x}_i \otimes \mathbf{z}_j)' \boldsymbol{\beta}_{xz}$$

LASSO, random effects, neural nets, ...

(Harris 2015, *Methods in Ecology and Evolution*)

(species by response, or sites by composition)

Latent variable models

also see Pledger & Arnold (2014, *J Comp Graph Stats*)

The question?



- Test (env impact)
- Composition
- Explain spp variation
- Predict (new sites or spp)
- Classify**
- Ordination
- Study correlations

Abundance
(mean-variance)

Multivariate
(high-dimensional)

row-resample GLMs

$$g(\mu_{ij}) = \alpha_i + \mathbf{x}'_i \boldsymbol{\beta}_j$$

Traits (\mathbf{z}_j) as predictors, interaction with \mathbf{x}_i :

$$g(\mu_{ij}) = \mathbf{x}'_i \boldsymbol{\beta}_x + \mathbf{z}'_j \boldsymbol{\beta}_z + (\mathbf{x}_i \otimes \mathbf{z}_j)' \boldsymbol{\beta}_{xz}$$

LASSO, random effects, neural nets, ...

(Harris 2015, *Methods in Ecology and Evolution*)

Finite mixture models

Pledger & Arnold (2014, *JCGS*), Dunstan *et al.* (2011 *Ecol Model...*)

Latent variable models

also see Pledger & Arnold (2014, *J Comp Graph Stats*)

The question?



- Test (env impact)
- Composition
- Explain spp variation
- Predict (new sites or spp)
- Classify
- Ordination
- Study correlations

Data properties?

Abundance
(mean-variance)

Multivariate
(high-dimensional)

row-resample GLMs

$$g(\mu_{ij}) = \alpha_i + \mathbf{x}'_i \boldsymbol{\beta}_j$$

Traits (\mathbf{z}_j) as predictors, interaction with \mathbf{x}_i :

$$g(\mu_{ij}) = \mathbf{x}'_i \boldsymbol{\beta}_x + \mathbf{z}'_j \boldsymbol{\beta}_z + (\mathbf{x}_i \otimes \mathbf{z}_j)' \boldsymbol{\beta}_{xz}$$

LASSO, random effects, neural nets, ...

(Harris 2015, *Methods in Ecology and Evolution*)

Finite mixture models

Pledger & Arnold (2014, *JCGS*), Dunstan *et al.* (2011 *Ecol Model...*)

Latent variable models

also see Pledger & Arnold (2014, *J Comp Graph Stats*)

The question?

- Test (env impact)
- Composition
- Explain spp variation
- Predict (new sites or spp)
- Classify
- Ordination
- Study correlations

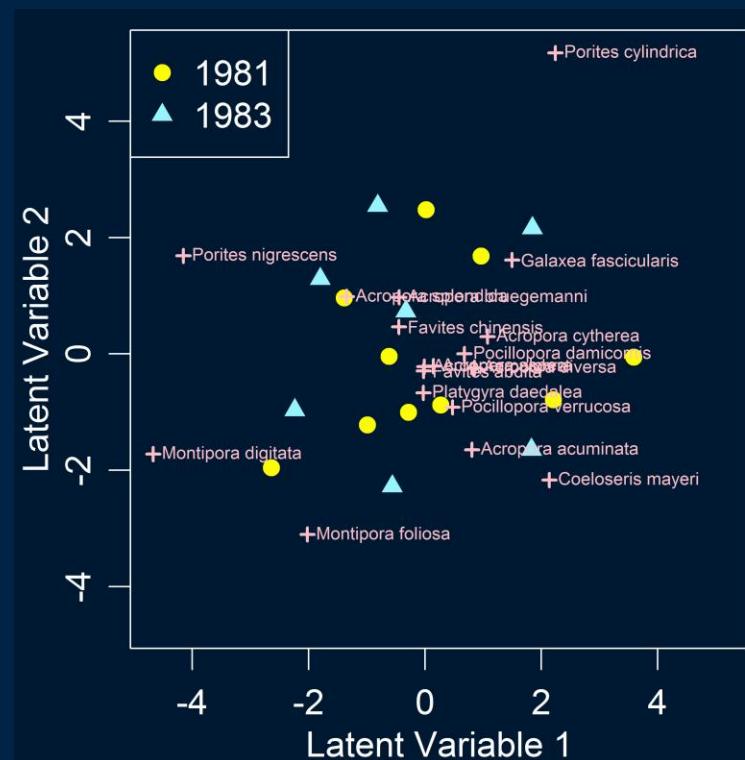


Data properties?

Abundance
(mean-variance)

Multivariate
(high-dimensional)

Can use a latent variable model (“biplot”):



Warton et al. (2015,
Trend Ecol Evol)

The question?



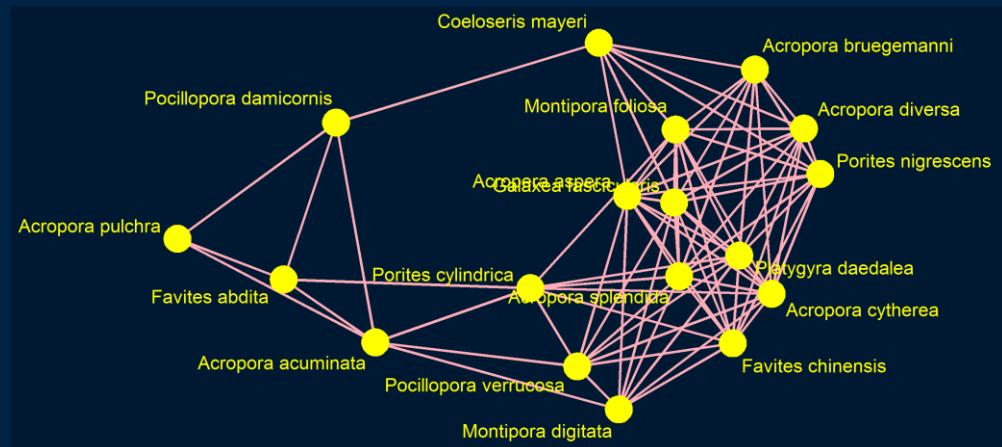
- Test (env impact)
- Composition
- Explain spp variation
- Predict (new sites or spp)
- Classify
- Ordination
- Study correlations

Abundance
(mean-variance)

Multivariate
(high-dimensional)

Or graphical modelling
 Σ^{-1} assumed to be sparse, estimate to maximise
 $\log \mathcal{L}(\beta, \Sigma; \mathbf{y}) - \lambda \|\Sigma^{-1}\|_1$

Popovic & Warton (in review)



The question?



- Test (env impact)
- Composition
- Explain spp variation
- Predict (new sites or spp)
- Classify
- Ordination
- Study correlations

Data properties?

Abundance
(mean-variance)
GLM+...

Multivariate
(high-dimensional)
copula

row-resample GLMs

$$g(\mu_{ij}) = \alpha_i + \mathbf{x}'_i \boldsymbol{\beta}_j$$

Traits (\mathbf{z}_j) as predictors, interaction with \mathbf{x}_i :

$$g(\mu_{ij}) = \mathbf{x}'_i \boldsymbol{\beta}_x + \mathbf{z}'_j \boldsymbol{\beta}_z + (\mathbf{x}_i \otimes \mathbf{z}_j)' \boldsymbol{\beta}_{xz}$$

LASSO, random effects, neural nets, ...

(Harris 2015, *Methods in Ecology and Evolution*)

Finite mixture models

Latent variable models

also see Pledger & Arnold (2014, *J Comp Graph Stats*)

Graphical models (and LVMs)

The question?



Data properties?

- Test (env impact)
- Composition
- Explain spp variation
- Predict (new sites or spp)
- Classify
- Ordination
- Study correlations

Abundance
(mean-variance)
GLM+...

Multivariate
(high-dimensional)
GEE, hierarchical, copula

row-resample GLMs

$$g(\mu_{ij}) = \alpha_i + \mathbf{x}'_i \boldsymbol{\beta}_j$$

Traits (\mathbf{z}_i) as predictors, interaction with \mathbf{x}_i :

$$g(\mu_{ij}) = \mathbf{x}'_i \boldsymbol{\beta}_x + \mathbf{z}'_j \boldsymbol{\beta}_z + (\mathbf{x}_i \otimes \mathbf{z}_j)' \boldsymbol{\beta}_{xz}$$

LASSO, random effects, neural nets, ...

(Harris 2015, *Methods in Ecology and Evolution*)

Finite mixture models

Latent variable models

also see Pledger & Arnold (2014, *J Comp Graph Stats*)

Graphical models (and LVMs)



Software for multivariate models

Test (env impact)	resample GLMs (anova.manyglm on mvabund)
Composition	$g(\mu_{ij}) = \alpha_i + \mathbf{x}'_i \boldsymbol{\beta}_j$ (manyany on mvabund)
Explain spp variation	Traits (\mathbf{z}_j) as predictors, interaction with \mathbf{x}_i : (traitglm on mvabund)
Predict (new sites or spp)	LASSO (glm1path on mvabund, or glmnet), random effects (lme4 etc), neural nets (mistnet)
Classify	Finite mixture models (speciesmix, RCPmod)
Ordination	Latent variable models (boral, HMSC)
Study correlations	Graphical models (and LVMs)

The case of the missing model: The modernisation of multivariate analysis in ecology

- Introduction
- The Bray-Curtis distance and other 1980's memorabilia
- Building a model for multivariate abundances in ecology
- A 2020 vision for multivariate analysis in ecology

A 2020 vision for multivariate analysis

Less of this:

$$d_{ii'} = \frac{\sum_j |y_{ij} - y_{i'j}|}{\sum_j (y_{ij} + y_{i'j})}$$

More of this:

$$g(\mu_{ij}) = \mathbf{x}'_i \boldsymbol{\beta}_x + \mathbf{z}'_j \boldsymbol{\beta}_z + (\mathbf{x}_i \otimes \mathbf{z}_j)' \boldsymbol{\beta}_{xz} + \mathbf{z}'_i \boldsymbol{\lambda}_j$$

$$\mathbf{z}_i \sim \mathcal{N}(0, 1)$$



To-do list for 2020:

Abundance
(mean-variance)

Multivariate
(high-dimensional)

Test (env impact)

Composition

Explain spp
variation

Predict
(new sites or spp)

Classify

Ordination

Study correlations

To-do list for 2020:

	Abundance (mean-variance)	Multivariate (high-dimensional)
Test (env impact)		
Composition		<ul style="list-style-type: none">• Model-based inference with many parameters
Explain spp variation		
Predict (new sites or spp)		
Classify		
Ordination		
Study correlations		

To-do list for 2020:

	Abundance (mean-variance)	Multivariate (high-dimensional)
Test (env impact)		
Composition		<ul style="list-style-type: none">• Model-based inference with many parameters
Explain spp variation		<ul style="list-style-type: none">• Classification + high-dimensional response
Predict (new sites or spp)		
Classify		
Ordination		
Study correlations		

To-do list for 2020:

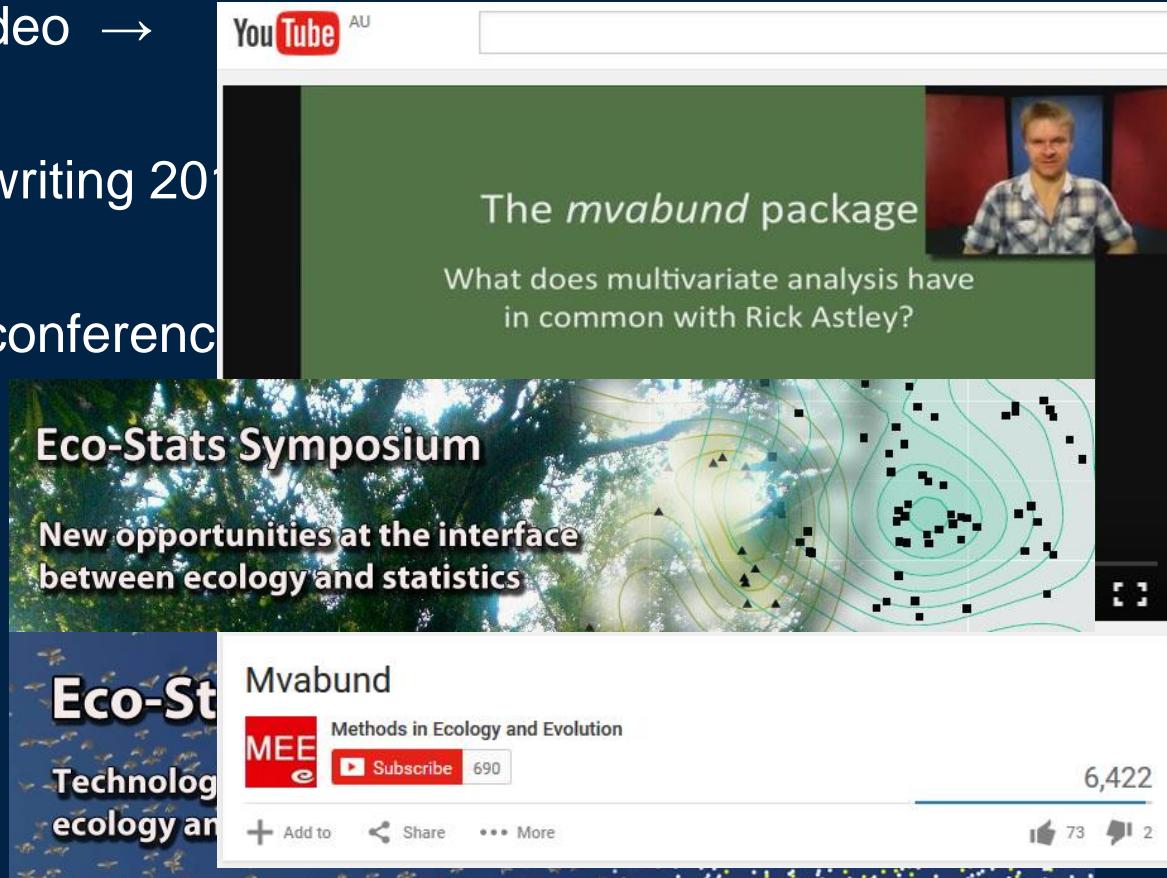
	Abundance (mean-variance)	Multivariate (high-dimensional)
Test (env impact)		
Composition		<ul style="list-style-type: none">• Model-based inference with many parameters
Explain spp variation		<ul style="list-style-type: none">• Classification + high-dimensional response
Predict (new sites or spp)		<ul style="list-style-type: none">• <i>Hierarchical classification using models</i>
Classify		
Ordination		
Study correlations		

To-do list for 2020:

	Abundance (mean-variance)	Multivariate (high-dimensional)
Test (env impact)		
Composition		<ul style="list-style-type: none">• Model-based inference with many parameters
Explain spp variation		<ul style="list-style-type: none">• Classification + high-dimensional response
Predict (new sites or spp)		<ul style="list-style-type: none">• <i>Hierarchical classification</i> using models• Spread the word to ecologists...
Classify		
Ordination		
Study correlations		

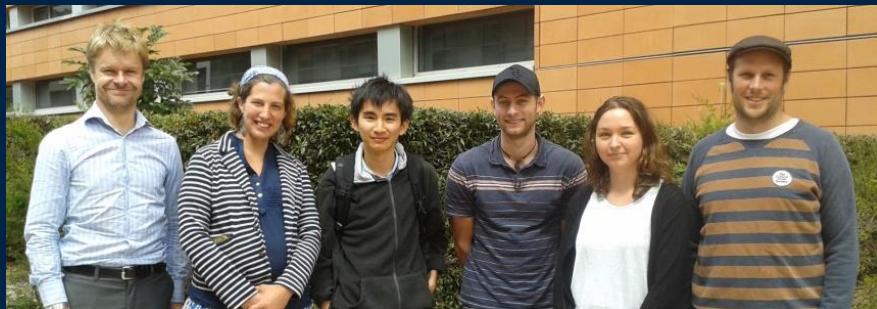
Spreading the word...

- Free well-documented software (`mvabund` etc)
- Youtube video →
- Blog posts
- Textbook (writing 2018)
- Workshops
- Eco-Stats conference



Acknowledgements

- UNSW Eco-Stats Group and Collaborators
- Australian Research Council
 - Future Fellowship scheme (FT120100501)
 - Discovery scheme (DP0985886, DP120100882, DP150100823)
- UNSW
 - School of Mathematics and Statistics
 - Evolution and Ecology Research Centre
- Ecologists who shared their data!



Multivariate abundance data

- The `manyglm` function
- Other model-based approaches?
- Post-hoc testing
- Closing advice on regression modelling

The `manyglm` function

The `manyglm` function in the `mvabund` package was designed for multivariate abundance data. How it deals with key data properties:

Multivariate: It uses row resampling for inference (as discussed in *Design-based inference*; first session today), to preserve the correlation between variables (species).

Abundance: `manyglm` fits a separate GLM to each species. Use the `family` argument to find a suitable model and mean-variance relationship for your data, and `plot` to check assumptions.

```
> library(mvabund)
> load("reveg.RData")
> revegMV=mvabund(reveg)
> ft.reveg=manyglm(revegMV~treatment+offset(log(pitfalls)),family="negative.binomial")
> anova(ft.reveg)
```

Time elapsed: 0 hr 0 min 9 sec

Analysis of Deviance Table

Model: manyglm(formula = revegMV ~ treatment + offset(log(pitfalls)),
Model: family = "negative.binomial")

Multivariate test:

	Res.Df	Df.diff	Dev	Pr(>Dev)
(Intercept)	9			
treatment	8	1	78.25	0.024 *

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Arguments:

Test statistics calculated assuming uncorrelated response (for faster computation)

P-value calculated using 999 resampling iterations via resampling (to account for corre

N.B. `manyglm` defaults to `family="negative.binomial"`

Computation time

Note that it took almost 10 second to run the `anova` function on a small dataset with 24 variables. Bigger datasets take minutes or hours!

The main problem is that **resampling is computationally intensive** – by default, this function will fit a `glm` to each variable 1000 times.

For larger datasets try setting `nBoot=50` or `100` to get a faster but more approximate answer. Then scale it up to `1000` when you need a final answer for publication.

N.B. You can also use the `show.time="all"` argument to get updates every 100 bootstrap samples, e.g. `anova(ft.revdeg,nBoot=500,show.time="all")`

manyglm test statistics

The default test statistic on `anova.manyglm` is a sum-of-LR statistic – it computes the likelihood ratio statistic separately for each taxon, then sums across taxa for an overall measure of effect of treatment on the invertebrate community.

By summing across taxa, the sum-of-LR statistic assumes independence of variables. **Hang on...** doesn't this ignore species correlation then?

While correlation is not accounted for in the test statistic, **it is accounted for in the *P*-value, by resampling rows of observations.**

So the procedure is valid even if the independence assumption used in constructing the test statistic is wrong.

Statistics that account for correlation

If you want to account for correlation between variables in the test statistic:

1) use the “score test statistic” or “Wald test statistic”. These are available via the `test` argument.

2) select the type of correlation you want. These are available via the `cor.type` argument. Options currently available include:

`cor.type="I"` (Default) Assumes independence for test statistic calculation – sums across taxa.

`cor.type="R"` Assumes correlation between all variables. Slower and is unstable if there are many variables compared to the number of observations.

`cor.type="shrink"` A good middle option between the previous two. Use this to account for correlation unless you have only a few variables.

```
> ft.reveg=manyglm(revegMV~treatment+offset(log(pitfalls)))
> anova(ft.reveg,test="wald",cor.type="shrink")
```

Time elapsed: 0 hr 0 min 6 sec

Analysis of Variance Table

```
Model: manyglm(formula = revegMV ~ treatment + offset(log(pitfalls)),
Model:       family = "negative.binomial")
```

Multivariate test:

	Res.Df	Df.diff	wald	Pr(>wald)
(Intercept)	9			
treatment	8	1	8.698	0.039 *

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Arguments:

Test statistics calculated assuming correlated response via ridge regularization
P-value calculated using 999 resampling iterations via resampling (to account for corre

The `manyglm` function - summary

You can also use the `summary` function for `manyglm` objects, but:

- as for `glm` objects, results aren't quite as trustworthy.
- it can take even longer to run, especially for complex models.

Model selection for `manyglm`

Many generic model selection functions work for `manyglm` objects – e.g. `AIC`, `predict`, `drop1`.

But model-based approaches (e.g. `AIC`) should be used with caution, because they do not take into account correlation between variables. That is why we prefer to use resampling (as something of a workaround).

Cross-validation is OK though – provided that you do the training/test **split by rows**.

Checking assumptions

Same assumptions as for GLMs, plus a correlation assumption:

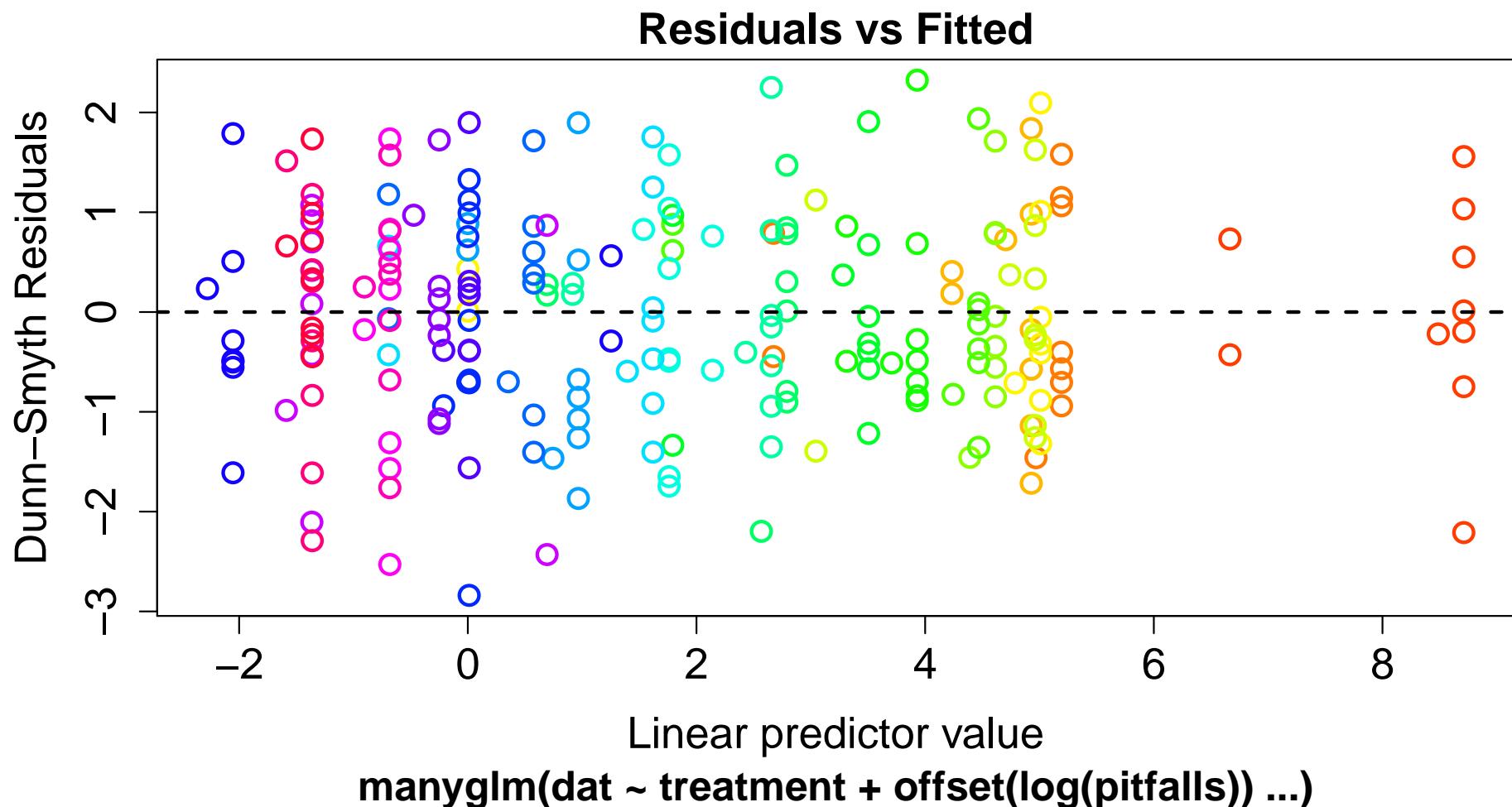
1. The observed y values are **independent**, after conditioning on x .
2. The y values come from **a known distribution** (from the exponential family) with known **mean-variance relationship** $V(\mu)$.
3. There is **straight line relationship** between some known function of the mean of y and each x

$$g(\mu_y) = \beta_0 + \mathbf{x}^T \boldsymbol{\beta}$$

4. Residuals have **constant correlation matrix** across observations.

Check assumptions 2-3 as for GLMs – using Dunn-Smyth residual plots.

```
> plot(ft.reveg)
```



Checking the mean-variance assumption

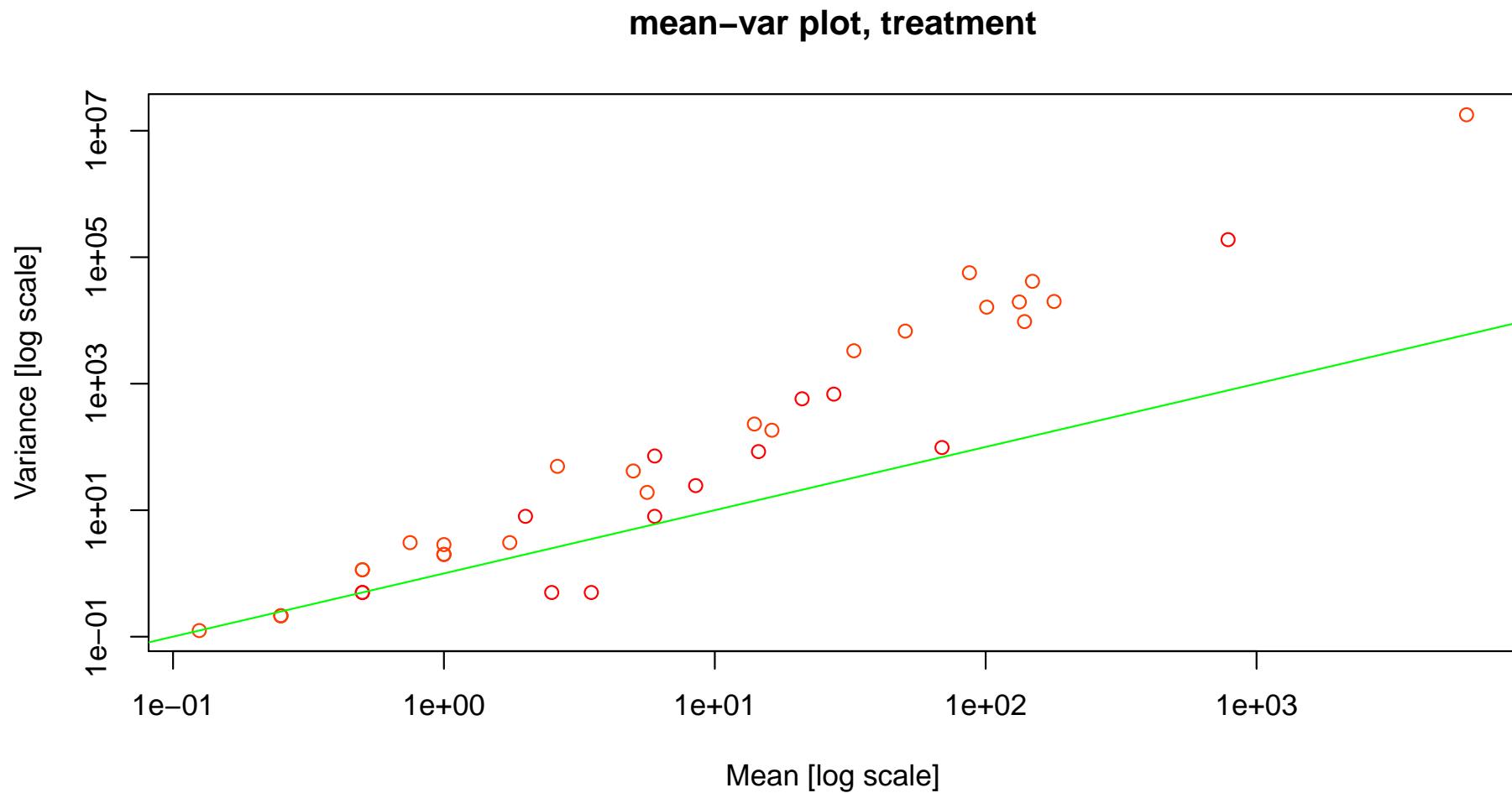
You can check the mean-variance assumption visually. Plot Dunn-Smyth residuals (plot for `manyglm` objects) and check for no fan-shaped pattern.

Alternatively, the `meanvar.plot` function constructs a mean-variance plot – **sample variances against sample means for different species** (and different treatment groups, if specified).

You can check the Poisson assumption by adding a `variance=mean` line (y -intercept=0, slope=1):

```
> meanvar.plot(revegMV~treatment)
> abline(a=0,b=1,col="green")
```

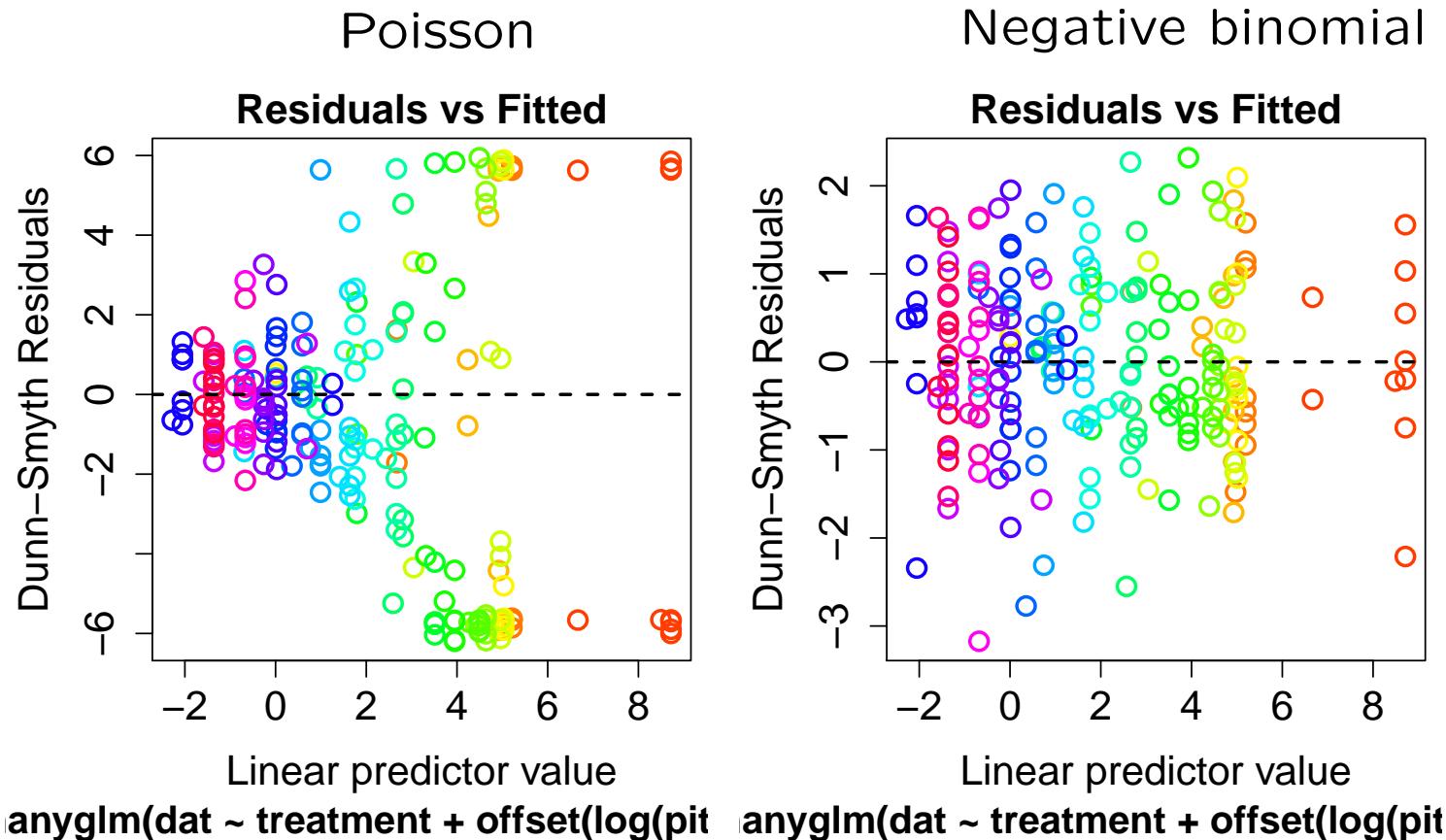
A mean-variance plot to check mean-variance assumption:



How's the Poisson assumption looking?

Residual plots to check mean-variance assumption:

```
> ft.revegP=manyglm(revegMV~treatment+offset(log(pitfalls)),family="poisson")
> plot(ft.revegP)
> ft.revegNB=manyglm(revegMV~treatment+offset(log(pitfalls)),family="negative.binomial")
> plot(ft.revegNB)
```



So which model should Anthony go with?

Other model-based approaches?

This is a fast-moving literature with a bunch of options out there.
Examples:

- `boral` or `HMSC` for model-based ordination (contact me for faster alternative code)
- `SpeciesMix` to classify species by environmental response
- `traitglm` on the `mvabund` package to including functional traits
- `manyany` on the `mvabund` package for more flexibility (and slower computation times).
- `saint` (contact Gordana) for model-based network models from multivariate abundance data
- `mistnet` for predictive modelling using neural networks

Post-hoc testing

Anthony has established there is an effect of bush regeneration on invertebrate communities:

```
> ft.reveg=manyglm(revegMV~treatment+offset(log(pitfalls)))  
> anova(ft.reveg)
```

Time elapsed: 0 hr 0 min 18 sec

...

Multivariate test:

	Res.Df	Df.diff	Dev	Pr(>Dev)
(Intercept)	9			
treatment	8	1	78.25	0.022 *

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

...

Now what?

In which variable(s) is there an effect?

The next obvious step is to think about which variables there is an effect in. This can be done by adding a p.uni argument:

```
> an = anova(ft.reveg,p.uni="adjusted")
> an
```

Analysis of Deviance Table

```
Model: manyglm(formula = dat ~ treatment + offset(log(pitfalls)), family = "negative.binomial")
```

Multivariate test:

	Res.Df	Df.diff	Dev	Pr(>Dev)
(Intercept)	9			
treatment	8	1	78.25	0.022 *
...				

In which variable(s) is there an effect?

...

Univariate Tests:

	Acarina		Amphipoda		Araneae		Blattodea	
	Dev	Pr(>Dev)	Dev	Pr(>Dev)	Dev	Pr(>Dev)	Dev	Pr(>Dev)
(Intercept)								
treatment	8.538	0.208	9.363	0.172	0.493	0.979	10.679	
	Coleoptera		Collembola		Dermoptera			
	Pr(>Dev)		Dev	Pr(>Dev)	Dev	Pr(>Dev)	Dev	
(Intercept)								
treatment	0.117	9.741	0.151	6.786	0.307	0.196		
	Diotocardia		Diplura		Diptera			
	Pr(>Dev)		Dev	Pr(>Dev)	Dev	Pr(>Dev)	Dev	Pr(>Dev)
(Intercept)								
treatment	0.979		0	0.984	2.24	0.850	5.93	0.380
	Formicidae		Haplotauxida		Hemiptera			
	Dev	Pr(>Dev)	Dev	Pr(>Dev)	Dev	Pr(>Dev)	Dev	Pr(>Dev)
(Intercept)								
treatment	0.831	0.979		2.889	0.782	1.302	0.965	
	Hymenoptera		Isopoda		Larvae		Lepidoptera	
	Dev	Pr(>Dev)	Dev	Pr(>Dev)	Dev	Pr(>Dev)	Dev	
(Intercept)								
treatment	4.254	0.561	1.096	0.979	0.463	0.979	0.913	
	Polydesmida		Pseudoscorpionida					
	Pr(>Dev)		Dev	Pr(>Dev)			Dev	Pr(>Dev)
(Intercept)								
treatment	0.979		1.451	0.960		1.056	0.979	
...								

In which variable(s) is there an effect?

`p.uni` allows univariate test statistics to be stored for each response variable. `p.uni="adjusted"` adjusts for **multiple testing** to control family-wise Type I error.

Note however that adjusted P -values are **very conservative**. So it is not uncommon to get global significance but no univariate significance – e.g. Anthony has good evidence of an effect on invertebrate communities, but can't point to any individual taxon as being significant.

In which variable(s) is there an effect?

However, you can still work out which are the “indicator” taxa which contributed most to the significant result – those with the **biggest test statistics**.

Univariate test stats are stored for an in an\$uni.test, and the second row gives test stats for the treatment effect. The “top 5”:

```
> s = sort(an$uni.test[2,],decreasing=T,index.return=T)
> s$x[1:5]
Blattodea Coleoptera Amphipoda Acarina Collembola
10.679374 9.741038 9.362519 8.537903 6.785946
```

The global test statistic, stored in an\$table[2,3], is 78.25. So the proportion of the difference in deviance due to the top 5 taxa is:

```
> sum(s$x[1:5])/an$table[2,3]
[1] 0.5764636
```

Parameter estimates/confidence intervals

The `confint` function hasn't been implemented (yet)...sorry

But you can get parameter estimates and their standard errors from `ft.revdeg$coef` and `ft.revdeg$stderr` respectively. For our top five taxa:

```
> coef(ft.revdeg)[,s$ix[1:5]]  
          Blattodea Coleoptera Amphipoda Acarina Collembola  
(Intercept) -0.3566749 -1.609438 -16.42495 1.064711 5.056246  
treatmentRevdeg -3.3068867 5.009950 19.42990 2.518570 2.045361  
> ft.revdeg$stderr[,s$ix[1:5]]  
          Blattodea Coleoptera Amphipoda Acarina Collembola  
(Intercept) 0.3779645 1.004969 707.1068 0.5171539 0.4879159  
treatmentRevdeg 1.0690450 1.066918 707.1069 0.5713194 0.5453801
```

Parameter estimates/confidence intervals

So for example the estimated effect of bush regeneration on beetles (*Coleoptera*) is an increase in $\log(\mu)$ by 5.0, with an approximate 95% CI being $5.0 \pm 2 \times 1.1 = 5.0 \pm 2.2$.

Closing advice

Which regression method do I use?

Assuming you are doing some sort of regression (relating y to x), the following steps might help you sort out what method to use:

- What's the research question of interest?
- What's the data, in particular:
 - How many response variables are there?
 - What type of response(s)?
 - Random factors, or all fixed?
 - Correlated observations?
 - Linear or non-linear effects?

What's the research question of interest?

The research question strongly determines whether you want to do:

- Descriptive statistics (no inference)
- Hypothesis testing
- Confidence intervals for key parameters
- Model selection

N.B. **Always** do some descriptive stats!

How many response variables?

- One – univariate analysis (e.g. linear models, GLMs, ...)
- A few – multivariate analysis (e.g. linear models to each species, `mvabund` with `cor.type="R"`)
- Heaps – multivariate analysis (e.g. `mvabund` with `cor.type="I"` and use design-based inference)

What type of response variable(s)?

- Continuous – try linear models (on possibly transformed response)
- Discrete – GLMs
- Presence/absence – GLMs
- Categorical (ordinal or nominal) – variations on GLMs (not discussed in this course)

Random or fixed factors? Correlated observations?

If random, consider if you want to include them as random effects.
If you have **correlated observations** with some special correlation structure (e.g. in space, time, or phylogeny), you can handle this by introducing random effects to induce correlation.

Including random effects makes things messier – especially for GLMs and multivariate data.

Linear or non-linear effects?

If you have enough data, you can fit a smoother to account for non-linearity, if required (“generalised additive models”, `mgcv` package).

What to do with a fancy new method

There are heaps of things we haven't talked about. High on my list:

- Spatial models (accounting for spatial autocorrelation)
- Longitudinal analysis, time series models, state-space models (for repeated measures)
- `smatr` – studying how size variables scale against each other.
- Phylogenetic regression (when species are your data but response is constrained by phylogeny)
- Point process models (when your data are point locations)

If we got through all of that, I would add more to the list too!

Some of these topics have brief introductory tutorials on the Eco-Stats blog. See link from the Eco-Stats homepage:

www.eco-stats.unsw.edu.au

What to do with a fancy new method

What steps should you go through in working out how to use some new method?

- Google it. Is there a text, tutorial, powerpoint slides on this? Ideally, by the developer of the software (or some other relatively serious name).
- If there is R software for it, does it come with a manual or vignette?
- What research questions is this new fandangle method used for? Is that the sort of thing you want to do?
- What sort of properties does my data need to have for this to be useful? How do I check this using my data?
- Road-test the software – go to the worked examples and run them yourself (On R, any code in the `Examples` section of a help file will run from the command line when the package has been loaded).
- Ask!