Tutorial 9.2a - Nested ANOVA

05 Apr 2018

If you are completely ontop of the conceptual issues pertaining to Nested ANOVA, and just need to use this tutorial in order to learn about Nested ANOVA in R, you are invited to skip down to the section on Nested ANOVA in R.

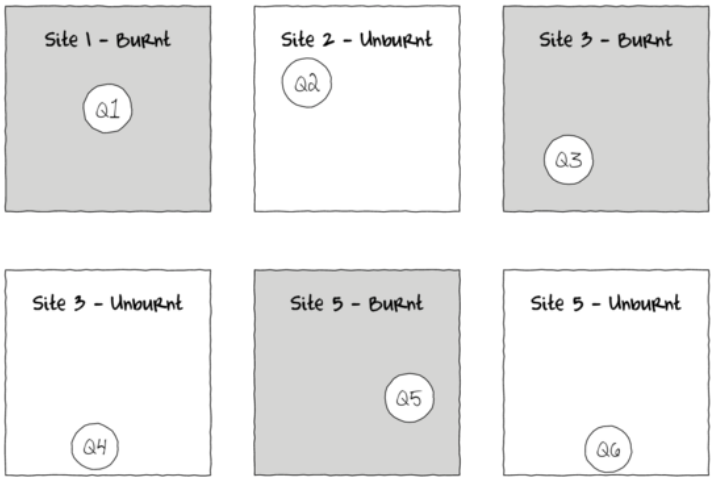
Section 1: Overview

When single sampling units are selected amongst highly heterogeneous conditions, it is unlikely that these single units will adequately represent the populations and repeated sampling is likely to yield very different outcomes.

In the depiction of a single factor ANOVA below, a single quadrat has been randomly placed in each of the six Sites. If the conditions within each site were very heterogeneous, then the exact location of the quadrat in the site is likely to be very important.

As a result, the amount of variation within the main treatment effect (unexplained variability or noise) remains high thereby potentially masking any detectable effects (the signal) due to the measured treatments. Although this problem can be addressed by increased replication (having more sites of each treatment type), this is not always practical or possible.

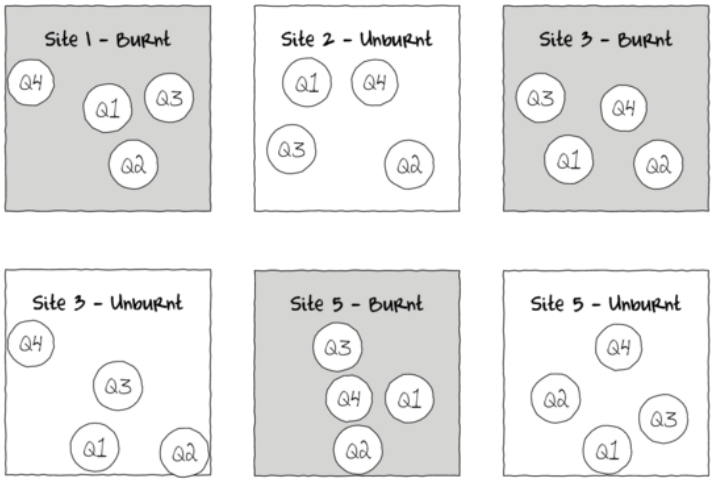
For example, if we were investigating the impacts of fuel reduction burning across a highly heterogeneous landscape, our ability to replicate adequately might be limited by the number of burn sites available.



Alternatively, sub-replicates within each of the sampling units (e.g.~sites) can be collected (and averaged) so as to provided better representatives for each of the units (see the figure below) and ultimately reduce the unexplained variability of the test of treatments.

In essence, the sub-replicates are the replicates of an additional *nested* factor whose levels are nested within the main treatment factor. A nested factor refers to a factor whose levels are unique within each level of the factor it is nested within and each level is only represented once.

For example, the fuel reduction burn study design could consist of three burnt sites and three un-burnt (control) sites each containing four quadrats (replicates of site and sub-replicates of the burn treatment). Each site represents a unique level of a random factor (any given site cannot be both burnt and un-burnt) that is nested within the fire treatment (burned or not).



A nested design can be thought of as a hierarchical arrangement of factors (hence the alternative name *hierarchical* designs) whereby a treatment is progressively sub-replicated.

As an additional example, imagine an experiment designed to comparing the leaf toughness of a number of tree species. Working down the hierarchy, five individual trees were randomly selected within (nested within) each species, three branches were randomly selected within each tree, two leaves were randomly selected within each branch and the force required to shear the leaf material in half (transversely) was measured in four random locations along the leaf. Clearly any given leaf can only be from a single branch, tree and species.

Each level of sub-replication is introduced to further reduce the amount of unexplained variation and thereby increasing the power of the test for the main treatment effect. Additionally, it is possible to investigate which scale has the greatest (or least, etc) degree of variability - the level of the species, individual tree, branch, leaf, leaf region etc.

Nested factors are typically random factors, of which the levels are randomly selected to represent all possible levels (e.g.~sites). When the main treatment effect (often referred to as Factor A) is a fixed factor, such designs are referred to as a *mixed model nested ANOVA*, whereas when Factor A is random, the design is referred to as a *Model II nested ANOVA*.

Fixed nested factors are also possible. For example, specific dates (corresponding to particular times during a season) could be nested within season. When all factors are fixed, the design is referred to as a *Model I mixed model*.

Fully nested designs (the topic of this chapter) differ from other multi-factor designs in that all factors within (below) the main treatment factor are nested and thus interactions are un-replicated and cannot be tested. Indeed, interaction effects (interaction between Factor A and site) are assumed to be zero.

Section 2: Linear models

The linear models for two and three factor nested design are:

$$y_{ijk} = \mu + \alpha_i + \beta_{j(i)} + \varepsilon_{ijk}$$

$$y_{ijkl} = \mu + \alpha_i + \beta_{j(i)} + \gamma_{k(j(i))} + \varepsilon_{ijkl}$$

where μ is the overall mean, α is the effect of Factor A, β is the effect of Factor B, γ is the effect of Factor C and ε is the random unexplained or residual component.

Section 3: Null hypotheses

Separate null hypotheses are associated with each of the factors, however, nested factors are typically only added to absorb some of the unexplained variability and thus, specific hypotheses tests associated with nested factors are of lesser biological importance. Hence, rather than estimate the effects of random effects, we instead estimate how much variability they contribute.

3.1: Factor A - the main treatment effect

3.1.1: Fixed

$H_0(A)$: $\mu_1 = \mu_2 = \ldots = \mu_i = \mu$ (the population group means are all equal)

The mean of population 1 is equal to that of population 2 and so on, and thus all population means are equal to an overall mean. If the effect of the i^{th} group is the difference between the i^{th} group mean and the overall mean ($\alpha_i = \mu_i - \mu$) then the H_0 can alternatively be written as:

$H_0(A)$: $\alpha_1 = \alpha_2 = \ldots = \alpha_i = 0$ (the effect of each group equals zero)

If one or more of the α_i are different from zero (the response mean for this treatment differs from the overall response mean), the null hypothesis is not true indicating that the treatment does affect the response variable.

3.1.2: Random

$H_0(A)$: $\sigma^2_\alpha = 0$ (population variance equals zero)

There is no added variance due to all possible levels of A.

3.2: Factor B - the nested factor

3.2.1: Random (typical case)

$H_0(B)$: $\sigma^2_\beta = 0$ (population variance equals zero)

There is no added variance due to all possible levels of B within the (set or all possible) levels of A.

3.2.2: Fixed

$H_0(B)$: $\mu_{1(1)} = \mu_{2(1)} = \ldots = \mu_{j(i)} = \mu$ (the population group means of B (within A) are all equal)

$H_0(B)$: $\beta_{1(1)} = \beta_{2(1)} = \ldots = \beta_{j(i)} = 0$ (the effect of each chosen B group equals zero)

The null hypotheses associated with additional factors, are treated similarly to Factor B above.

Section 4: Analysis of variance

Analysis of variance sequentially partitions the total variability in the response variable into components explained by each of the factors (starting with the factors lowest down in the hierarchy - the most deeply nested) and the components unexplained by each factor. Explained variability is calculated by subtracting the amount unexplained by the factor from the amount unexplained by a reduced model that does not contain the factor.

When the null hypothesis for a factor is true (no effect or added variability), the ratio of explained and unexplained components for that factor (*F*-ratio) should follow a theoretical *F*-distribution with an expected value less than 1. The appropriate unexplained residuals and therefore the appropriate *F*-ratios for each factor differ according to the different null hypotheses associated with different combinations of fixed and random factors in a nested linear model (see Table below).

			A fixed/random, B random		**A fixed/random, B fixed**	
Factor	**d.f**	**MS**	**F-ratio**	**Var. comp.**	**F-ratio**	**Var.comp.**
A	$a-1$	MS_A	$\frac{MS_A}{MS_{B'(A)}}$	$\frac{MS_A - MS_{B'(A)}}{nb}$	$\frac{MS_A}{MS_{Resid}}$	$\frac{MS_A - MS_{Resid}}{nb}$
B′(A)	$(b-1)a$	$MS_{B'(A)}$	$\frac{MS_{B'(A)}}{MS_{Resid}}$	$\frac{MS_{B'(A)} - MS_{Resid}}{n}$	$\frac{MS_{B'(A)}}{MS_{Resid}}$	$\frac{MS_{B'(A)} - MS_{Resid}}{n}$
Residual (=$N'(B'(A))$)	$(n-1)ba$	MS_{Resid}				
		A fixed/random, B random				

	Unbalanced		`summary(aov(y~A+Error(B), data))`
			`library(nlme)`
			`VarCorr(lme(y~A,random=1 |B, data))`			

	Unbalanced		`library(nlme)` `anova(lme(y~A,random=1 |B, data), type='marginal')`			
		A fixed/random, B fixed				
	Unbalanced		`summary(aov(y~A+B, data))`			

			`contrasts(data$B) <- contr.sum`
			`library(car)`
			`Anova(aov(y~A/B, data), type='III')`			

Section 5: Variance components

As previously alluded to, it can often be useful to determine the relative contribution (to explaining the unexplained variability) of each of the factors as this provides insights into the variability at each different scales.

These contributions are known as *Variance components* and are estimates of the added variances due to each of the factors. For consistency with leading texts on this topic, I have included estimated variance components for various balanced nested ANOVA designs in the above table. However, variance components based on a modified version of the maximum likelihood iterative model fitting procedure (REML) is generally recommended as this accommodates both balanced and unbalanced designs.

While there are no numerical differences in the calculations of variance components for fixed and random factors, fixed factors are interpreted very differently and arguably have little biological meaning (other to infer relative contribution). For fixed factors, variance components estimate the variance between the means of the specific populations that are represented by the selected levels of the factor and therefore represent somewhat arbitrary and artificial populations. For random factors, variance components estimate the variance between means of all possible populations that could have been selected and thus represents the true population variance.

Section 6: Assumptions

An *F*-distribution represents the relative frequencies of all the possible *F*-ratio's when a given null hypothesis is true and certain assumptions about the residuals (denominator in the *F*-ratio calculation) hold.

Consequently, it is also important that diagnostics associated with a particular hypothesis test reflect the denominator for the appropriate *F*-ratio. For example, when testing the null hypothesis that there is no effect of Factor A ($H_0(A) : \alpha_i = 0$) in a mixed nested ANOVA, the means of each level of Factor B are used as the replicates of Factor A.

As with single factor anova, hypothesis testing for nested ANOVA assumes the residuals are:

- normally distributed. Factors higher up in the hierarchy of a nested model are based on means (or means of means) of lower factors and thus the Central Limit Theory would predict that normality will usually be satisfied for the higher level factors. Nevertheless, boxplots using the appropriate scale of replication should be used to explore normality. Scale transformations are

often useful.

- equally varied. Boxplots and plots of means against variance (using the appropriate scale of replication) should be used to explore the spread of values. Residual plots should reveal no patterns (see Figure~??). Scale transformations are often useful.
- independent of one another - this requires special consideration so as to ensure that the scale at which sub-replicates are measured is still great enough to enable observations to be independent.

Section 7: Pooling denominator terms

Designs that incorporate fixed and random factors (either nested or factorial), involve *F*-ratio calculations in which the denominators are themselves random factors other than the overall residuals. Many statisticians argue that when such denominators are themselves not statistically significant (at the 0.25 level), there are substantial power benefits from pooling together successive non-significant denominator terms. Thus an *F*-ratio for a particular factor might be recalculated after pooling together its original denominator with its denominators denominator and so on.

The conservative 0.25 is used instead of the usual 0.05 to reduce further the likelihood of Type II errors (falsely concluding an effect is non-significant - that might result from insufficient power).

Section 8: Unbalanced nested designs

For a simple completely balanced nested ANOVA, it is possible to pool together (calculate their mean) each of the sub-replicates within each nest (=site) and then perform single factor ANOVA on those aggregates. Indeed, for a balanced design, the estimates and hypothesis for Factor A will be identical to that produced via nested ANOVA.

However, if there are an unequal number of sub-replicates within each nest, then the single factor ANOVA will be less powerful that a proper nested ANOVA.

Unbalanced designs are those designs in which sample (subsample) sizes for each level of one or more factors differ. These situations are relatively common in biological research, however such imbalance has some important implications for nested designs.

- Firstly, hypothesis tests are more robust to the assumptions of normality and equal variance when the design is balanced.
- Secondly (and arguably, more importantly), the model contrasts are not orthogonal (independent) and the sums of squares component attributed to each of the model terms cannot be calculated by simple additive partitioning of the total sums of squares.

In such situations, exact *F*-ratios cannot be constructed (at least in theory), variance components calculations are more complicated and significance tests cannot be computed. The denominator MS in an \textit{F}-ratio is determined by examining the expected value of the mean squares of each term in a model. Unequal sample sizes result in expected means squares for which there are no obvious logical comparators that enable the impact of an individual model term to be isolated.

The severity of this issue depends on which scale of the sub-sampling hierarchy the unbalance(s) occurs as well whether the unbalance occurs in the replication of a fixed or random factor. For example, whilst unequal levels of the first nesting factor (e.g. unequal number of burn vs un-burnt sites) has no effect on *F*-ratio construction or hypothesis testing for the top level factor (irrespective of whether either of the factors are fixed or random), unequal sub-sampling (replication) at the level of a random (but not fixed) nesting factor will impact on the ability to construct *F*-ratios and variance components of all terms above it in the hierarchy.

There are a number of alternative ways of dealing with unbalanced nested designs. All alternatives assume that the imbalance is not a direct result of the treatments themselves. Such outcomes are more appropriately analysed by modelling the counts of surviving observations via frequency analysis.

- Split the analysis up into separate smaller simple ANOVA's each using the means of the nesting factor to reflect the appropriate scale of replication. As the resulting sums of squares components are thereby based on an aggregated dataset the analyses then inherit the procedures and requirements of single ANOVA.
- Adopt mixed-modelling techniques (see below).

Section 9: Linear mixed effects models

Although the term `mixed-effects' can be used to refer to any design that incorporates both *fixed* and *random* predictors, its use is more commonly restricted to designs in which factors are nested or grouped within other factors. Typical examples include nested, longitudinal (measurements repeated over time) data, repeated measures and blocking designs.

Furthermore, rather than basing parameter estimations on observed and expected mean squares or error strata (as outline above), mixed-effects models estimate parameters via maximum likelihood (ML) or residual maximum likelihood (REML). In so doing, mixed-effects models more appropriately handle estimation of parameters, effects and variance components of unbalanced designs (particularly for random effects).

Resulting fitted (or expected) values of each level of a factor (for example, the expected population site means) are referred to as Best Linear Unbiased Predictors (BLUP's). As an acknowledgement that most estimated site means will be more extreme than the underlying true population means they estimate (based on the principle that smaller sample sizes result in greater chances of more extreme observations and that nested sub-replicates are also likely to be highly intercorrelated), BLUP's are less spread from the overall mean than are simple site means. In addition, mixed-effects models naturally model the 'within-block' correlation structure that complicates many longitudinal designs.

Whilst the basic concepts of mixed-effects models have been around for a long time, recent computing advances and adoptions have greatly boosted the popularity of these procedures.

Linear mixed effects models are currently at the forefront of statistical development, and as such, are very much a work in progress - both in theory and in practice. Recent developments have seen a further shift away from the traditional practices associated with degrees of freedom, probability distribution and p-value calculations.

The traditional approach to inference testing is to compare the fit of an alternative (full) model to a null (reduced) model (via an *F*-ratio). When assumptions of normality and homogeneity of variance apply, the degrees of freedom are easily computed and the *F*-ratio has an exact *F*-distribution to which it can be compared.

However, this approach introduces two additional problematic assumptions when estimating fixed effects in a mixed effects model. Firstly, when estimating the effects of one factor, the parameter estimates associated with other factor(s) are assumed to be the

true values of those parameters (not estimates). Whilst this assumption is reasonable when all factors are fixed, as random factors are selected such that they represent one possible set of levels drawn from an entire population of possible levels for the random factor, it is unlikely that the associated parameter estimates accurately reflect the true values. Consequently, there is not necessarily an appropriate *F*-distribution.

Furthermore, determining the appropriate degrees of freedom (nominally, the number of independent observations on which estimates are based) for models that incorporate a hierarchical structure is only possible under very specific circumstances (such as completely balanced designs).

Degrees of freedom is a somewhat arbitrary defined concept used primarily to select a theoretical probability distribution on which a statistic can be compared. Arguably, however, it is a concept that is overly simplistic for complex hierarchical designs.

Most statistical applications continue to provide the `approximate' solutions (as did earlier versions within R). However, R linear mixed effects development leaders argue strenuously that given the above shortcomings, such approximations are variably inappropriate and are thus omitted.

9.1: MCMC sampling

Markov chain Monte Carlo (MCMC) sampling methods provide a Bayesian-like alternative for inference testing. Markov chains use the mixed model parameter estimates to generate posterior probability distributions of each parameter from which Monte Carlo sampling methods draw a large set of parameter samples.

These parameter samples can then be used to calculate highest posterior density (HPD) intervals (also known as Bayesian credible intervals). Such intervals indicate the interval in which there is a specified probability (typically 95%) that the true population parameter lies. Furthermore, whilst technically against the spirit of the Bayesian philosophy, it is also possible to generate P values on which to base inferences.

Section 10: Nested ANOVA in R

Routine	lme	lmer	glmmTMB
fit model	nlme::lme(y~A, random=~1|B, data)	lme4::lmer(y~x+(1|B), data)	glmmTMB::glmmTMB(y~x+(1|B), data)
raw residuals	residuals(mod)	residuals(mod)	residuals(mod)
standardized residuals	residuals(mod, type='pearson')	residuals(mod, type='pearson')	residuals(mod, type='pearson')
normalized residuals	rstandard(mod)	residuals(mod, type='pearson', scaled=TRUE)	

10.1: Scenario and Data

Imagine we has designed an experiment in which we intend to measure a response (y) to one of treatments (three levels; 'a1', 'a2' and 'a3'). The treatments occur at a spatial scale (over an area) that far exceeds the logistical scale of sampling units (it would take too long to sample at the scale at which the treatments were applied). The treatments occurred at the scale of hectares whereas it was only feasible to sample y using 1m quadrats.

Given that the treatments were naturally occurring (such as soil type), it was not possible to have more than five sites of each treatment type, yet prior experience suggested that the sites in which you intended to sample were very uneven and patchy with respect to y.

In an attempt to account for this inter-site variability (and thus maximize the power of the test for the effect of treatment, you decided to employ a nested design in which 10 quadrats were randomly located within each of the five replicate sites per three treatments. As this section is mainly about the generation of artificial data (and not specifically about what to do with the data), understanding the actual details are optional and can be safely skipped. Consequently, I have folded (toggled) this section away.

Details of data generation

10.2: Exploratory data analysis

10.2.1: Normality and homogeneity of variance

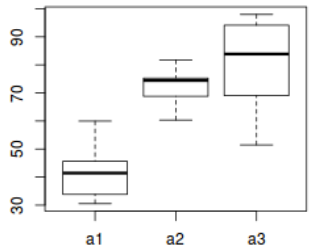
Recall that in the nested design the observations are collected at a sub-replicate rather than the replicate level. That is, the actual replicates of the main treatment effect are the blocks not the quadrats. Therefore, when assessing assumptions pertaining to the residuals, it is necessary to first aggregate the observations to the blocks (=sites) - use the average of quadrats within each block as the replicates for the blocks.

```
library(tidyverse)
# calculate the site means
data.nest.agg = data.nest %>% group_by(A, Sites) %>% summarize(y = mean(y))
data.nest.agg
```

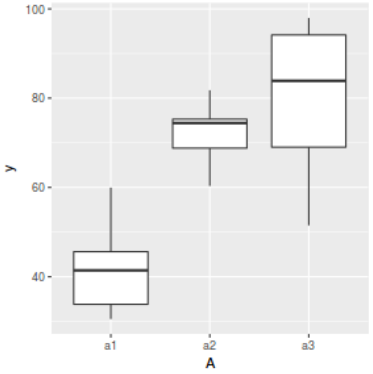
```
# A tibble: 15 x 3
# Groups:   A [?]
   A     Sites     y
   <fct> <fct> <dbl>
 1 a1    S1     33.8
 2 a1    S2     41.4
 3 a1    S3     30.5
 4 a1    S4     60.0
 5 a1    S5     45.6
 6 a2    S6     60.3
 7 a2    S7     75.3
 8 a2    S8     81.8
 9 a2    S9     74.4
10 a2    S10    68.8
11 a3    S11    98.0
```

```
12 a3    S12    83.9
13 a3    S13    69.0
14 a3    S14    51.5
15 a3    S15    94.1
```

```
boxplot(y ~ A, data.nest.agg)
```




```
ggplot(data.nest.agg, aes(y = y, x = A)) + geom_boxplot()
```



Conclusions: no obvious violations of non-normality or homogeneity of variance. Note that assessing normality can be a little difficult from such small numbers of replicates (5 sites per treatment).

10.2.2: Design balance

```
replications(y ~ A + Error(Sites), data.nest)
 A
50
```

```
# OR
replications(y ~ A + Sites, data.nest)
    A Sites
   50    10
```

Conclusions: the design is balanced and therefore type I (sequential) sums of squares are appropriate (if required).

10.3: Model fitting or statistical analysis

There are numerous ways of fitting a nested ANOVA in R.

lme | lmer | glmmTMB | Traditional aov

Linear mixed effects modelling via the `lme()` *function*. This method is one of the original implementations in which separate variance-covariance matrices are incorporated into a interactive sequence of (generalized least squares) and maximum likelihood (actually REML) estimates of 'fixed' and 'random effects'.

Rather than fit just a single, simple random intercepts model, it is common to fit other related alternative models and explore which model fits the data best. For example, we could also fit a random intercepts and slope model. We could also explore other variance-covariance structures (autocorrelation or heterogeneity).

```
library(nlme)
# random intercept
data.nest.lme <- lme(y ~ A, random = ~1  | Sites, data.nest, method = "REML")
# random intercept/slope
data.nest.lme1 <- lme(y ~ A, random = ~A | Sites, data.nest, method = "REML")
anova(data.nest.lme, data.nest.lme1)
               Model df      AIC      BIC    logLik   Test  L.Ratio p-value
data.nest.lme      1  5 927.7266 942.6788 -458.8633
data.nest.lme1     2 10 934.7325 964.6369 -457.3663 1 vs 2 2.994092  0.7009
```

Conclusions: the more complex random intercepts and slopes model does not fit the data significantly better than the simpler random intercepts model and thus the latter model will be used.

10.4: Model evaluation

10.4.1: Residuals

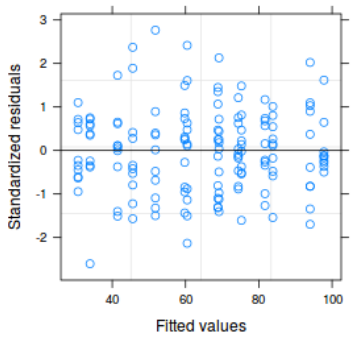
As always, exploring the residuals can reveal issues of heteroscadacity, non-linearity and potential issues with autocorrelation. Note for `lme()` and `lmer()` residual plots use standardized (normalized) residuals rather than raw residuals as the former reflect changes to the variance-covariance matrix whereas the later do not.

The following function will be used for the production of some of the qqnormal plots.

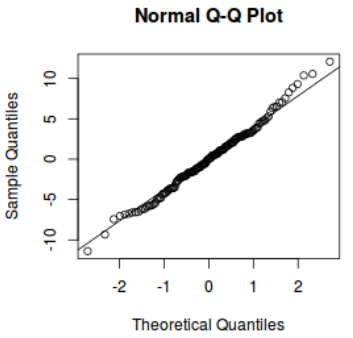
```
qq.line = function(x) {
    # following four lines from base R's qqline()
    y <- quantile(x[!is.na(x)], c(0.25 , 0.75 ))
    x <- qnorm(c(0.25 , 0.75 ))
    slope <- diff(y)/diff(x)
    int <- y[1L ] - slope * x[1L ]
    return(c(int = int, slope = slope))
}
```

lme | lmer | glmmTMB | Traditional aov

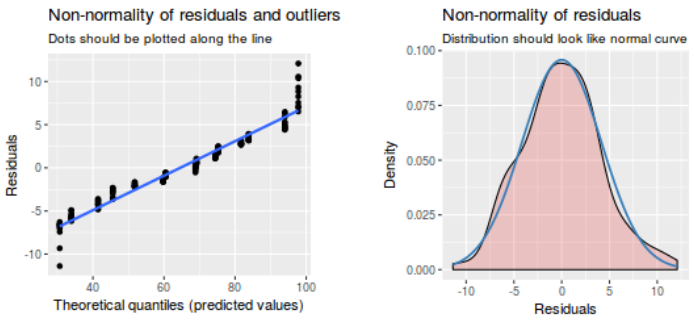
```
plot(data.nest.lme)
```

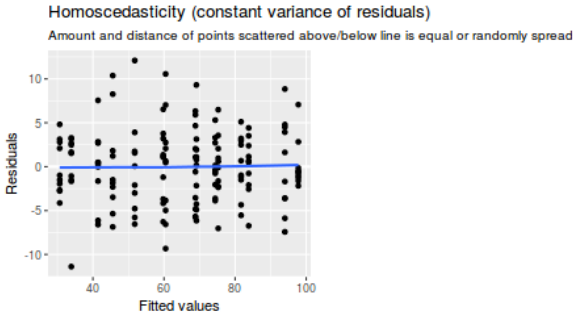



```
qqnorm(resid(data.nest.lme))
qqline(resid(data.nest.lme))
```




```
library(sjPlot)
plot_grid(plot_model(data.nest.lme, type = "diag"))
```





Conclusions: there are no issues obvious from the residuals.

10.5: Exploring model parameters

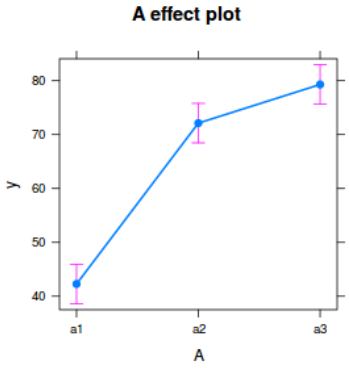
If there was any evidence that the assumptions had been violated, then we would need to reconsider the model and start the process again. In this case, there is no evidence that the test will be unreliable so we can proceed to explore the test statistics. As I had elected to illustrate multiple techniques for analysing this nested design, I will also deal with the summaries etc separately.

10.5.1: Partial effects plots

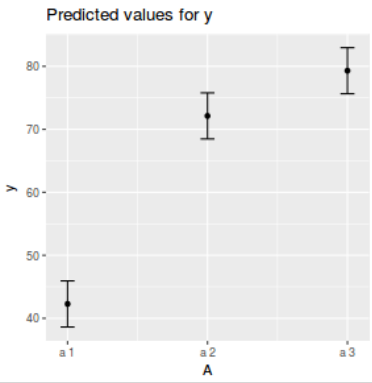
It is often useful to visualize partial effects plots while exploring the parameter estimates. Having a graphical representation of the partial effects typically makes it a lot easier to interpret the parameter estimates and inferences.

lme | lmer | glmmTMB

```
library(effects)
plot(allEffects(data.nest.lme))
```




```
library(sjPlot)
plot_model(data.nest.lme, type = "eff", terms = "A")
```



lme | lmer | glmmTMB | Traditional aov

```
summary(data.nest.lme)
```

```
Linear mixed-effects model fit by REML
 Data: data.nest
       AIC      BIC    logLik
  927.7266 942.6788 -458.8633

Random effects:
 Formula: ~1 | Sites
        (Intercept) Residual
StdDev:     13.6582 4.372252

Fixed effects: y ~ A
               Value Std.Error  DF  t-value p-value
(Intercept) 42.27936  6.139350 135 6.886618  0.0000
Aa2         29.84692  8.682352  12 3.437654  0.0049
Aa3         37.02026  8.682352  12 4.263851  0.0011
 Correlation:
    (Intr) Aa2
Aa2 -0.707
Aa3 -0.707  0.500

Standardized Within-Group Residuals:
         Min           Q1          Med           Q3          Max
-2.603787242 -0.572951701  0.004953998  0.620914933  2.765601716

Number of Observations: 150
Number of Groups: 15
```

```
intervals(data.nest.lme)
```

```
Approximate 95% confidence intervals

 Fixed effects:
               lower     est.    upper
(Intercept) 30.13761 42.27936 54.42110
Aa2         10.92970 29.84692 48.76414
Aa3         18.10304 37.02026 55.93748
attr(,"label")
[1] "Fixed effects:"

 Random Effects:
  Level: Sites
                   lower    est.    upper
sd((Intercept)) 9.117188 13.6582 20.46096

 Within-group standard error:
   lower     est.    upper
3.880632 4.372252 4.926153
```

```
anova(data.nest.lme)
```

```
            numDF denDF  F-value p-value
(Intercept)     1   135 331.8308  <.0001
A               2    12  10.2268  0.0026
```

```
library(broom)
tidy(data.nest.lme, effects = "fixed")
```

```
         term estimate std.error statistic      p.value
1 (Intercept) 42.27936  6.139350  6.886618 1.968597e-10
2          Aa2 29.84692  8.682352  3.437654 4.915711e-03
3          Aa3 37.02026  8.682352  4.263851 1.099991e-03
```

```
glance(data.nest.lme)
```

```
     sigma    logLik      AIC      BIC deviance
1 4.372252 -458.8633 927.7266 942.6788       NA
```

Conclusions:

- there is a significant effect of factor A
- more specifically, level a2 and a3 are both significantly higher than level a1
- sites accounted for a substantial amount of variability
- if we required to compare each group with each other group via multiple pairwise comparisons (whilst controlling for type I error rates), we might also conclude that there was no evidence that groups a2 and a3 differed from one another.

10.5.2: Variance components

lme | lmer | glmmTMB

```
VarCorr(data.nest.lme)
Sites = pdLogChol(1)
            Variance  StdDev
(Intercept) 186.54644 13.658200
Residual     19.11659  4.372252
```

10.5.3: R^2 approximations

Whilst R^2 is a popular goodness of fit metric in simple linear models, its use is rarely extended to (generalized) linear mixed effects models. The reasons for this include:

- there are numerous ways that R^2 could be defined for mixed effects models, some of which can result in values that are either difficult to interpret or illogical (for example negative R^2).
- perhaps as a consequence, software implementation is also largely lacking.

Nakagawa and Schielzeth (2013) discuss the issues associated with R^2 calculations and suggest a series of simple calculations to yield sensible R^2 values from mixed effects models.

An R^2 value quantifies the proportion of variance explained by a model (or by terms in a model) - the higher the value, the better the model (or term) fit. Nakagawa and Schielzeth (2013) offered up two R^2 for mixed effects models:

- **Marginal** R^2 - the proportion of total variance explained by the fixed effects.

$$\text{Marginal } R^2 = \frac{\sigma_f^2}{\sigma_f^2 + \sum_l^z \sigma_l^2 + \sigma_d^2 + \sigma_e^2}$$

  where σ_f^2 is the variance of the fitted values (i.e. $\sigma_f^2 = var(\mathbf{X}\beta)$) on the link scale, $\sum_l^z \sigma_l^2$ is the sum of the z random effects (including the residuals) and σ_d^2 and σ_e^2 are additional variance components appropriate when using non-Gaussian distributions.

- **Conditional** R^2 - the proportion of the total variance collectively explained by the fixed and random factors

$$\text{Conditional } R^2 = \frac{\sigma_f^2 + \sum_l^z \sigma_l^2}{\sigma_f^2 + \sum_l^z \sigma_l^2 + \sigma_d^2 + \sigma_e^2}$$

Since this tutorial is concerned with linear mixed effects models (and thus Gaussian distributions), we can ignore the σ_d^2 and σ_e^2 terms for now and return to them in Tutorial 11.2a.

lme | lmer | glmmTMB

```
library(MuMIn)
r.squaredGLMM(data.nest.lme)
      R2m       R2c
0.5571090 0.9588328

library(sjstats)
r2(data.nest.lme)
    R-squared: 0.9595
Omega-squared: 0.9595
```

The fixed effect of A (within Block) accounts for approximately 55.71% of the total variation in Y. The random effect of Block accounts for approximately 40.17% of the total variation in Y and collectively, the hierarchical level of Block (containing the fixed effect) explains approximately 95.88% of the total variation in Y.

10.6: Graphical summary

It is relatively trivial to produce a summary figure based on the raw data. Arguably a more satisfying figure would be one based on the modelled data.

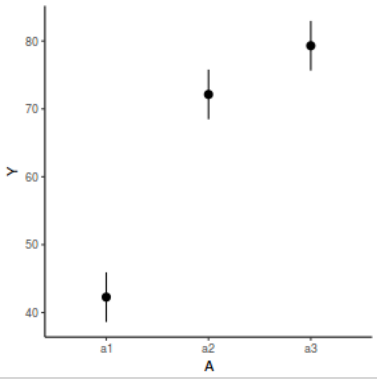
lme | lmer | glmmTMB

The new effects package seems to have a bug when calculating effects

```
library(effects)
data.nest.eff = as.data.frame(allEffects(data.nest.lme)[[1 ]])
```

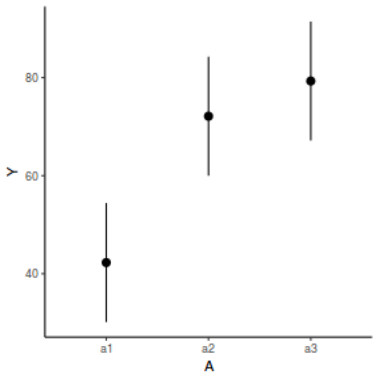
```
# OR
data.nest.eff = as.data.frame(Effect("A", data.nest.lme))

ggplot(data.nest.eff, aes(y = fit, x = A)) + geom_pointrange(aes(ymin = lower,
    ymax = upper)) + scale_y_continuous("Y") + theme_classic()
```



Or manually

```
newdata = data.frame(A = levels(data.nest$A))
Xmat = model.matrix(~A, data = newdata)
coefs = fixef(data.nest.lme)
fit = as.vector(coefs %*% t(Xmat))
se = sqrt(diag(Xmat %*% vcov(data.nest.lme) %*% t(Xmat)))
q = qt(0.975 , df = nrow(data.nest.lme$data) - length(coefs) -
    2 )
newdata = cbind(newdata, fit = fit, lower = fit - q * se, upper = fit +
    q * se)
ggplot(newdata, aes(y = fit, x = A)) + geom_pointrange(aes(ymin = lower,
    ymax = upper)) + scale_y_continuous("Y") + theme_classic()
```



Section 11: Another data set and scenario

Lets now add an additional hierarchical layer - sampling units within the quadrats.

Details of data generation

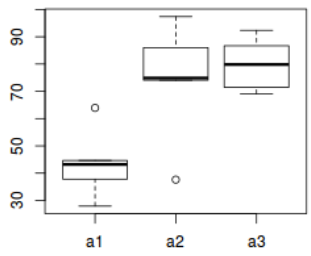
11.1: Exploratory data analysis

```
library(plyr)
# calculate the site means
data.nest1.agg <- ddply(data.nest1, ~A + Sites, function(x) {
    data.frame(y = mean(x$y))
})
data.nest1.agg
```

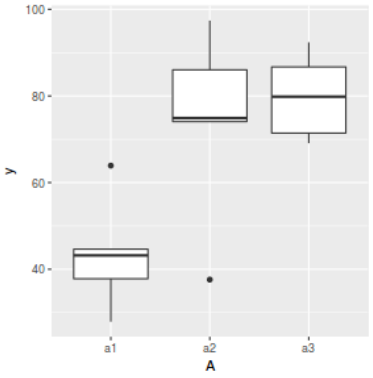
```
    A Sites        y
1  a1    S1 37.80088
2  a1    S4 63.91939
3  a1    S7 43.19601
4  a1   S10 44.63361
5  a1   S13 27.90133
6  a2    S2 74.88463
7  a2    S5 74.08441
8  a2    S8 86.03434
9  a2   S11 97.40098
10 a2   S14 37.61487
11 a3    S3 71.43908
12 a3    S6 69.10660
13 a3    S9 79.84349
```

```
14 a3   S12 86.70191
15 a3   S15 92.37056
```

```
boxplot(y ~ A, data.nest1.agg)
```




```
ggplot(data.nest1.agg, aes(y = y, x = A)) + geom_boxplot()
```




```
replications(y ~ A + Error(Sites/Quads), data.nest1)
```

```
 A
50
```

```
# OR
replications(y ~ A + Sites + Quads, data.nest1)
```

```
    A Sites Quads
   50    10    30
```

Conclusions: there are no immediately obvious issues with the data (notwithstanding the difficulties of exploring normality and homogeneity from small sample sizes) and the design is balanced and therefore type I (sequential) sums of squares are appropriate

11.2: Model fitting or statistical analysis

lme | lmer | glmmTMB | Traditional aov

Linear mixed effects modelling via the `lme()` *function*. This method is one of the original implementations in which separate variance-covariance matrices are incorporated into a interactive sequence of (generalized least squares) and maximum likelihood (actually REML) estimates of 'fixed' and 'random effects'.

Rather than fit just a single, simple random intercepts model, it is common to fit other related alternative models and explore which model fits the data best. For example, we could also fit a random intercepts and slope model. We could also explore other variance-covariance structures (autocorrelation or heterogeneity).

```
library(nlme)
data.nest1.lme <- lme(y ~ A, random = ~1  | Sites/Quads, data.nest1, method = "REML")
```

Conclusions: the more complex random intercepts and slopes model does not fit the data significantly better than the simpler random intercepts model and thus the latter model will be used.

11.3: Model evaluation

11.3.1: Residuals

As always, exploring the residuals can reveal issues of heteroscadacity, non-linearity and potential issues with autocorrelation. Note for `lme()` and `lmer()` residual plots use standardized (normalized) residuals rather than raw residuals as the former reflect changes to the variance-covariance matrix whereas the later do not.

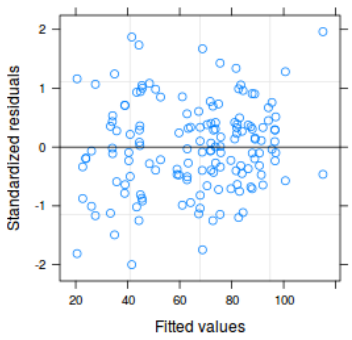
The following function will be used for the production of some of the qqnormal plots.

```
qq.line = function(x) {
    # following four lines from base R's qqline()
```

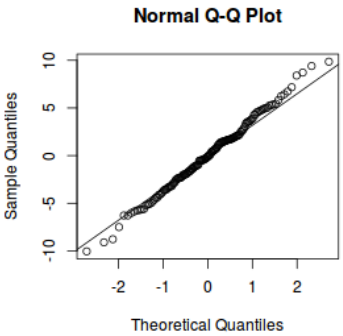
```
    y <- quantile(x[!is.na(x)], c(0.25 , 0.75 ))
    x <- qnorm(c(0.25 , 0.75 ))
    slope <- diff(y)/diff(x)
    int <- y[1L ] - slope * x[1L ]
    return(c(int = int, slope = slope))
}
```

lme | lmer | glmmTMB | Traditional aov

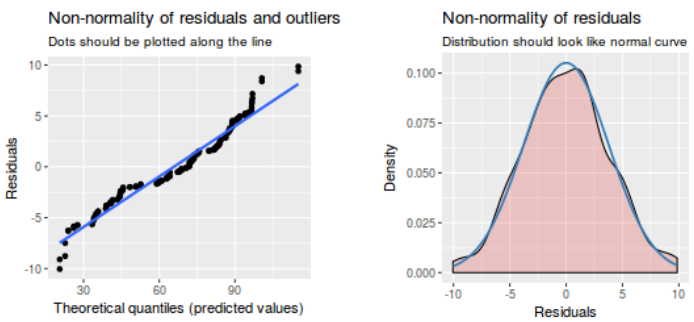
```
plot(data.nest1.lme)
```

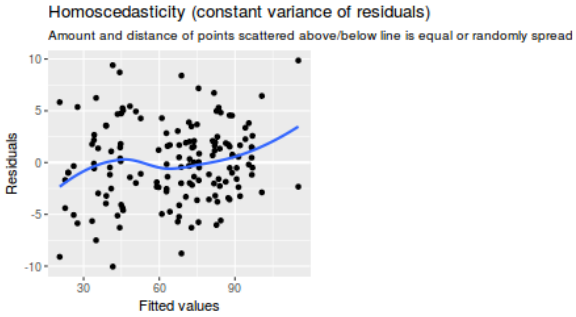



```
qqnorm(resid(data.nest1.lme))
qqline(resid(data.nest1.lme))
```




```
library(sjPlot)
plot_grid(plot_model(data.nest1.lme, type = "diag"))
```





Conclusions: there are no issues obvious from the residuals.

11.4: Exploring model parameters

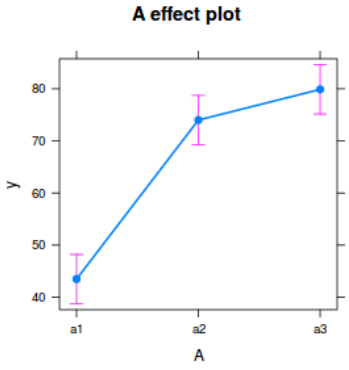
If there was any evidence that the assumptions had been violated, then we would need to reconsider the model and start the process again. In this case, there is no evidence that the test will be unreliable so we can proceed to explore the test statistics. As I had elected to illustrate multiple techniques for analysing this nested design, I will also deal with the summaries etc separately.

11.4.1: Partial effects plots

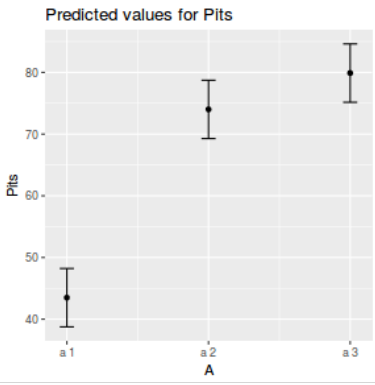
It is often useful to visualize partial effects plots while exploring the parameter estimates. Having a graphical representation of the partial effects typically makes it a lot easier to interpret the parameter estimates and inferences.

lme | lmer | glmmTMB

```
library(effects)
plot(allEffects(data.nest1.lme))
```




```
library(sjPlot)
plot_model(data.nest1.lme, type = "eff", terms = "A")
```



lme lmer glmmTMB Traditional aov

```
summary(data.nest1.lme)
```

```
Linear mixed-effects model fit by REML
 Data: data.nest1
       AIC      BIC    logLik
  1079.258 1097.201 -533.6292

Random effects:
 Formula: ~1 | Sites
        (Intercept)
StdDev:    15.59142

 Formula: ~1 | Quads %in% Sites
        (Intercept) Residual
StdDev:    8.003042 5.023277

Fixed effects: y ~ A
               Value Std.Error DF  t-value p-value
(Intercept) 43.49024  7.189234 75 6.049357  0.0000
Aa2         30.51360 10.167113 12 3.001206  0.0110
Aa3         36.40209 10.167113 12 3.580376  0.0038
 Correlation:
    (Intr) Aa2
Aa2 -0.707
Aa3 -0.707  0.500

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-2.00020166 -0.47356371 -0.02493572  0.41710101  1.96210317

Number of Observations: 150
Number of Groups:
          Sites Quads %in% Sites
             15               75
```

```
intervals(data.nest1.lme)
```

```
Approximate 95% confidence intervals

 Fixed effects:
                lower     est.    upper
(Intercept) 29.168554 43.49024 57.81193
Aa2          8.361367 30.51360 52.66584
Aa3         14.249850 36.40209 58.55432
attr(,"label")
[1] "Fixed effects:"

 Random Effects:
  Level: Sites
                   lower     est.    upper
sd((Intercept)) 10.18847 15.59142 23.85955
  Level: Quads
                   lower     est.    upper
sd((Intercept)) 6.445288 8.003042 9.937288

 Within-group standard error:
   lower     est.    upper
4.280419 5.023277 5.895056
```

```
anova(data.nest1.lme)
```

```
            numDF denDF   F-value p-value
(Intercept)     1    75 251.27426  <.0001
```

```
A              2    12   7.38726  0.0081
```

```
library(broom)
tidy(data.nest1.lme, effects = "fixed")
```

```
         term estimate std.error statistic      p.value
1 (Intercept) 43.49024  7.189234  6.049357 5.276461e-08
2          Aa2 30.51360 10.167113  3.001206 1.104191e-02
3          Aa3 36.40209 10.167113  3.580376 3.779659e-03
```

```
glance(data.nest1.lme)
```

```
     sigma    logLik      AIC      BIC deviance
1 5.023277 -533.6292 1079.258 1097.201       NA
```

Conclusions:

- there is a significant effect of factor A
- more specifically, level a2 and a3 are both significantly higher than level a1
- sites accounted for a substantial amount of variability
- if we required to compare each group with each other group via multiple pairwise comparisons (whilst controlling for type I error rates), we might also conclude that there was no evidence that groups a2 and a3 differed from one another.

11.4.2: Variance components

lme | lmer | glmmTMB

```
VarCorr(data.nest1.lme)
```

```
            Variance     StdDev
Sites =     pdLogChol(1)
(Intercept) 243.09238    15.591420
Quads =     pdLogChol(1)
(Intercept)  64.04868     8.003042
Residual     25.23331     5.023277
```

11.4.3: R^2 approximations

Whilst R^2 is a popular goodness of fit metric in simple linear models, its use is rarely extended to (generalized) linear mixed effects models. The reasons for this include:

- there are numerous ways that R^2 could be defined for mixed effects models, some of which can result in values that are either difficult to interpret or illogical (for example negative R^2).
- perhaps as a consequence, software implementation is also largely lacking.

Nakagawa and Schielzeth (2013) discuss the issues associated with R^2 calculations and suggest a series of simple calculations to yield sensible R^2 values from mixed effects models.

An R^2 value quantifies the proportion of variance explained by a model (or by terms in a model) - the higher the value, the better the model (or term) fit. Nakagawa and Schielzeth (2013) offered up two R^2 for mixed effects models:

- **Marginal** R^2 - the proportion of total variance explained by the fixed effects.

$$\text{Marginal } R^2 = \frac{\sigma_f^2}{\sigma_f^2 + \sum_l^z \sigma_l^2 + \sigma_d^2 + \sigma_e^2}$$

  where σ_f^2 is the variance of the fitted values (i.e. $\sigma_f^2 = var(\mathbf{X}\beta)$) on the link scale, $\sum_l^z \sigma_l^2$ is the sum of the z random effects (including the residuals) and σ_d^2 and σ_e^2 are additional variance components appropriate when using non-Gaussian distributions.

- **Conditional** R^2 - the proportion of the total variance collectively explained by the fixed and random factors

$$\text{Conditional } R^2 = \frac{\sigma_f^2 + \sum_l^z \sigma_l^2}{\sigma_f^2 + \sum_l^z \sigma_l^2 + \sigma_d^2 + \sigma_e^2}$$

Since this tutorial is concerned with linear mixed effects models (and thus Gaussian distributions), we can ignore the σ_d^2 and σ_e^2 terms for now and return to them in Tutorial 11.2a.

lme | lmer | glmmTMB

```
library(MuMIn)
r.squaredGLMM(data.nest1.lme)
```

```
      R2m       R2c
0.4353361 0.9571316
```

```
library(sjstats)
r2(data.nest1.lme)
```

```
    R-squared: 0.9737
Omega-squared: 0.9733
```

The fixed effect of A (within Block) accounts for approximately 43.53% of the total variation in Y. The random effect of Block accounts for approximately 52.18% of the total variation in Y and collectively, the hierarchical level of Block (containing the fixed effect) explains approximately 95.71% of the total variation in Y.

11.5: Graphical summary

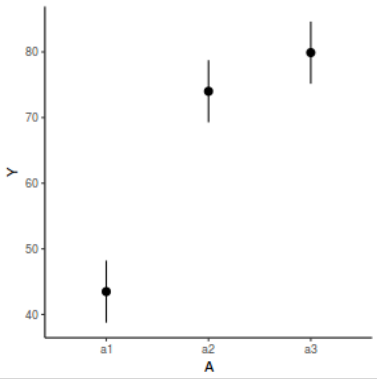
It is relatively trivial to produce a summary figure based on the raw data. Arguably a more satisfying figure would be one based on the modelled data.

lme | lmer | glmmTMB

The new effects package seems to have a bug when calculating effects

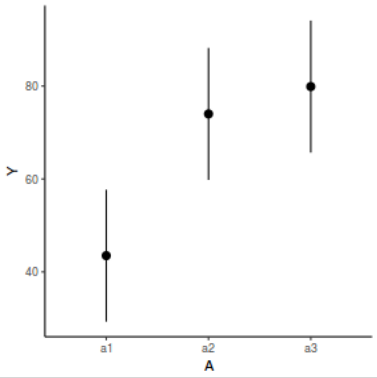
```
library(effects)
data.nest1.eff = as.data.frame(allEffects(data.nest1.lme)[[1 ]])
# OR
data.nest1.eff = as.data.frame(Effect("A", data.nest1.lme))

ggplot(data.nest1.eff, aes(y = fit, x = A)) + geom_pointrange(aes(ymin = lower,
    ymax = upper)) + scale_y_continuous("Y") + theme_classic()
```



Or manually

```
newdata = data.frame(A = levels(data.nest1$A))
Xmat = model.matrix(~A, data = newdata)
coefs = fixef(data.nest1.lme)
fit = as.vector(coefs %*% t(Xmat))
se = sqrt(diag(Xmat %*% vcov(data.nest1.lme) %*% t(Xmat)))
q = qt(0.975 , df = nrow(data.nest1.lme$data) - length(coefs) -
    2 )
newdata = cbind(newdata, fit = fit, lower = fit - q * se, upper = fit +
    q * se)
ggplot(newdata, aes(y = fit, x = A)) + geom_pointrange(aes(ymin = lower,
    ymax = upper)) + scale_y_continuous("Y") + theme_classic()
```



Section 12: References

Nakagawa, S. and H. Schielzeth (2013). "A general and simple method for obtaining R2 from generalized linear mixed-effects models". In: Methods in Ecology and Evolution 4.2, pp. 133–142. ISSN: 2041-210X. DOI: 10.1111/j.2041-210x.2012.00261.x. URL: http://dx.doi.org/10.1111/j.2041-210x.2012.00261.x.

Section 13: Worked Examples

Nested ANOVA references

- Logan (2010) - Chpt 12-14
- Quinn & Keough (2002) - Chpt 9-11

Example 1: Nested ANOVA - one between factor

In an unusually detailed preparation for an Environmental Effects Statement for a proposed discharge of dairy wastes into the Curdies River, in western Victoria, a team of stream ecologists wanted to describe the basic patterns of variation in a stream invertebrate thought to be sensitive to nutrient enrichment. As an indicator species, they focused on a small flatworm, Dugesia, and started by sampling populations of this worm at a range of scales. They sampled in two seasons, representing different flow regimes of the river - Winter and Summer. Within each season, they sampled three randomly chosen (well, haphazardly, because sites are nearly always chosen to be close to road access) sites. A total of six sites in all were visited, 3 in each season. At each site, they sampled six stones, and counted the number of flatworms on each stone.

Download Curdies data set

Format of curdies.csv data files

SEASON	SITE	DUGESIA	S4DUGESIA
WINTER	1	0.648	0.897
..	..	..	..
WINTER	2	1.016	1.004
..	..	..	..
WINTER	3	0.689	0.991
..	..	..	..
SUMMER	4	0	0
..	..	..	..

Each row represents a different stone

SEASON	Season in which flatworms were counted - fixed factor
SITE	Site from where flatworms were counted - nested within SEASON (random factor)
DUGESIA	Number of flatworms counted on a particular stone
S4DUGESIA	4th root transformation of DUGESIA variable

The hierarchical nature of this design can be appreciated when we examine an illustration of the spatial and temporal scale of the measurements.

- Within each of the two Seasons, there were three separate Sites (Sites not repeatidly measured across Seasons).
- Within each of the Sites, six logs were selected (haphazardly) from which the number of flatworms were counted.

So the Logs (smallest sampling units) are the replicates for the Sites (six reps per Site) and the Site means are the replicates for the two Seasons (three replicates within each Season).

Open the curdies data file.

Show code

Q1.1. The SITE variable is supposed to represent a random factorial variable (which site). However, because the contents of this variable are numbers, R initially treats them as numbers, and therefore considers the variable to be numeric rather than categorical. In order to force R to treat this variable as a factor (categorical) it is necessary to first convert this numeric variable into a factor (HINT)
.

Show code

Notice the data set - each of the nested factors is labelled differently - there can be no replicate for the random (nesting) factor.

Q1.2. What are the main hypotheses being tested?

a. H_0 Effect 1:

b. H_0 Effect 2:

Q1.3. In the table below, list the assumptions of nested ANOVA along with how violations of each assumption are diagnosed and/or the risks of violations are minimized.

Assumption	Diagnostic/Risk Minimization
I.	
II.	
III.	

Q1.4. **Check these assumptions** (HINT).

Show code

Note that for the effects of SEASON (Factor A in a nested model) there are only three values for each of the two season types. Therefore, boxplots are of limited value! Is there however, any evidence of violations of the assumptions (HINT)?

Show code

Show ggplot code

(Y or N)

If so, assess whether a transformation will address the violations (HINT) and then make the appropriate corrections.

Show code

Note, this tutorial is focussed on linear mixed effects against a Gaussian and therefore we will normalized our response (via forth-root transformation). Given that the data are counts. Arguably, it would be more appropriate to model the raw data against a poisson distribution. We will do so in Tutorial 11.2a

Q1.5. For each of the tests, state which error (or residual) term (state as Mean Square) will be used as the nominator and denominator to calculate the F ratio. Also include degrees of freedom associated with each term. Note this is a somewhat outdated procedure as there is far from consensus about the most appropriate estimate of denominator degrees of freedom. Nevertheless, it sometimes helps to visualize the model hierarchy this way.

Effect	Nominator (Mean Sq, df)	Denominator (Mean Sq, df)
SEASON		
SITE		

Q1.6. If there is no evidence of violations, fit the model;
S4DUGES = SEASON + SITE + CONSTANT
using a nested ANOVA (HINT). Fill (HINT) out the table below, make sure that you have treated SITE as a random factor when compiling the overall results.

Show traditional code

Show lme code

Show lmer code

Show glmmTMB code

Q1.7. Check the model diagnostics

- Residual plot

Show traditional code

Show lme code

Show lmer code

Show glmmTMB code

Q1.8. For each of the tests, state which error (or residual) term (state as Mean Square) will be used as the nominator and denominator to calculate the F ratio. Also include degrees of freedom associated with each term.

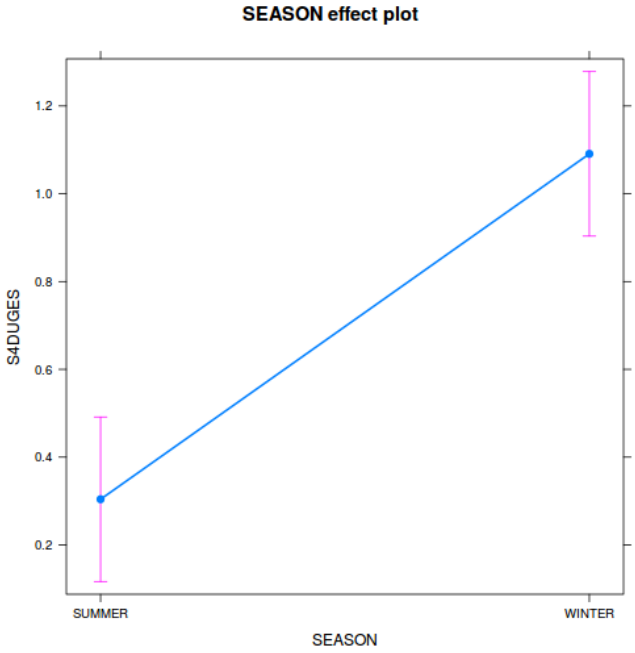
Source of variation	df	Mean Sq	F-ratio	P-value
SEASON				
SITE				
Residuals				

Estimate	Mean	Lower 95% CI	Upper 95% CI
Summer			
Effect size (Winter-Summer)			

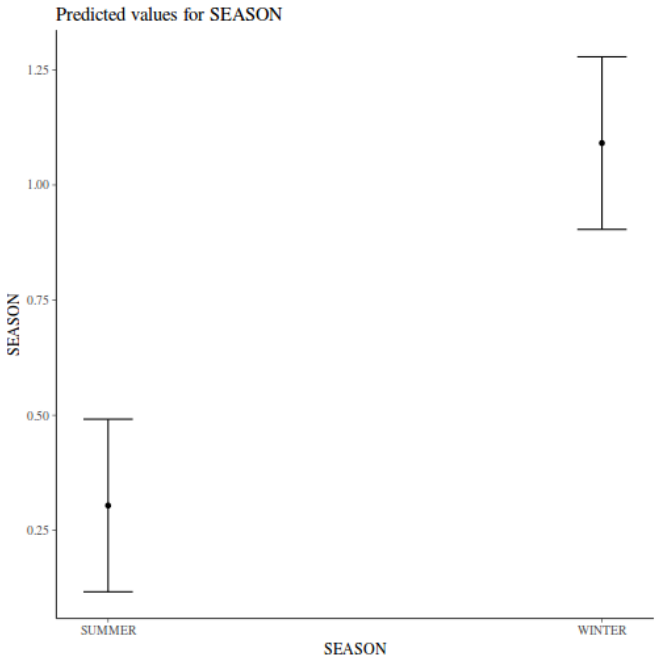
Q1.9. Lets produce a quick partial (marginal) effects plot to represent the model.

Show lme code

```
library(effects)
plot(allEffects(curdies.lme))
```



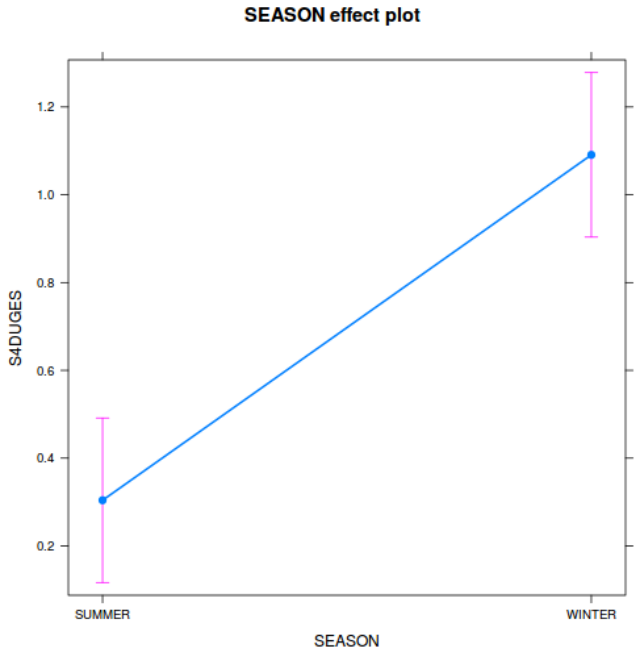

```
library(sjPlot)
plot_model(curdies.lme, type = "eff", terms = "SEASON")
```



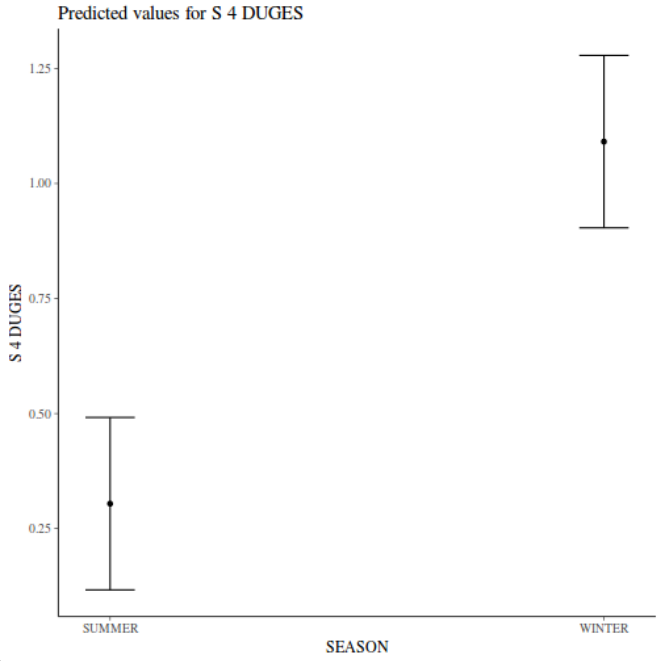
Show lmer code

```
library(effects)
```

```
plot(allEffects(curdies.lmer))
```

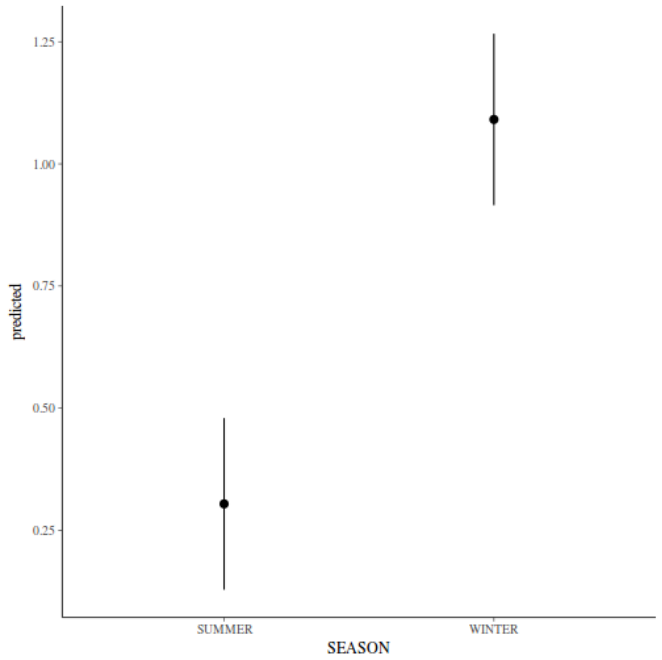



```
library(sjPlot)
plot_model(curdies.lmer, type = "eff", terms = "SEASON")
```

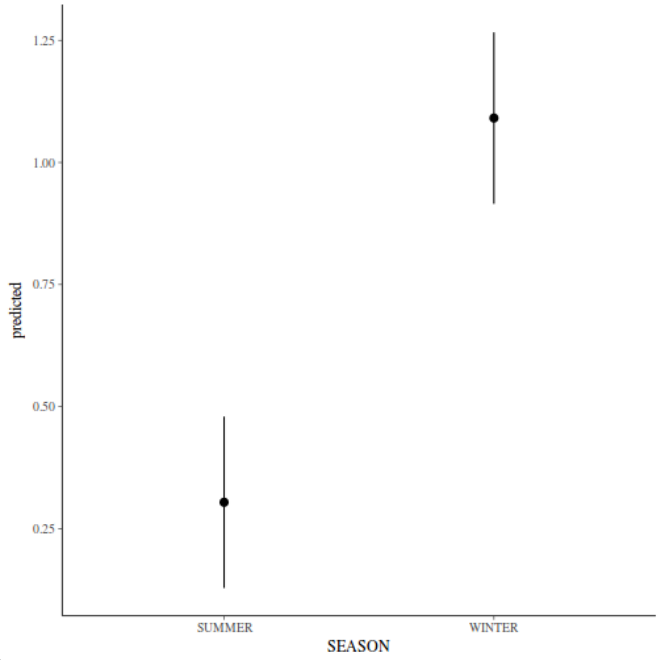


Show glmmTMB code

```
library(ggeffects)
# observation level effects averaged across margins
p = ggaverage(curdies.glmmTMB, terms = "SEASON")
p = cbind(p, SEASON = levels(curdies$SEASON))
ggplot(p, aes(y = predicted, x = SEASON)) + geom_pointrange(aes(ymin = conf.low, ymax = conf.high))
```




```
# marginal effects
p = ggpredict(curdies.glmmTMB, terms = "SEASON")
p = cbind(p, SEASON = levels(curdies$SEASON))
ggplot(p, aes(y = predicted, x = SEASON)) + geom_pointrange(aes(ymin = conf.low, ymax = conf.high))
```



Q1.10. Where is the major variation in numbers of flatworms? Between (seasons, sites or stones)?

Show code for lme

Show code for lme

Show code for glmmTMB

Q1.11. Finally, construct an appropriate summary figure to accompany the above analyses. Note that this should use the correct replicates for depicting error.

Show lme code

Show lmer code

Show glmmTMB code