Linear mixed-effects models

James S. Santangelo

Lesson preamble

Learning objectives

- Understand the limitations of standard linear models (e.g. linear regression, ANOVA).
- Understand the benefits of mixed-effects modelling.
- Understand the differences between fixed and random effects.
- Apply random intercept and random intercept and slope models to nested experimental data.

Lesson outline

Total lesson time: 2 hours

- Load and familiarize ourselves with the RIKZ data that will be used for the majority of today's lesson (10 min)
- Perform standard linear regression on a subset of the RIKZ data and check assumptions of model (i.e. recap from last week, 15 min)
- Explore in greater detail violation of an important assumption of standard linear models; namely, the independence of observations.
  - Explore ways to overcome this violation without the use of mixed-effects models. What are the drawbacks of these approaches? (15 min)
  - Apply both random intercept, random intercept and slope models and simple random-effects only models to RIKZ data. What are the differences between these models? Interpret model output. (40 min)
  - Using mixed-effects models for more deeply nested data. Difference between nested and crossed random effects (20 min)
- Discussion of fixed vs random effects, ML vs. REML, other types of mixed-effects models (e.g. GAM models, alternative variance structures, etc.) (20 min)

Setup

- `install.packages('plyr')`
- `install.packages('dplyr')` (or `tidyverse`)
- `install.packages('ggplot2')` (or `tidyverse`)
- `install.packages('broom')` (or `tidyverse`)
- `install.packages('lme4')`
- `install.packages('lmerTest')`
- `install.packages('ggforce')`
- `install.packages('convaveman')`

Quick linear model recap and extension

Last week we discussed how to apply linear models to data (e.g. linear regression, ANOVA, etc.) to understand the relationship between predictor (i.e. independent) and response (i.e. dependent) variables. While such models are incredibly powerful, they are underlain by numerous assumptions that should be met prior to placing any confidence in their results. As a reminder, these assumptions are:

1. Normality at each X value (or of the residuals)
2. Homogeneity of variances at each X
3. Fixed X
4. Independence of observations
5. Correct model specification

Typically, small amounts of non-normality and heterogeneity of variances (AKA heteroscedasticity) is alright and will not strongly bias the results. However, serious heterogeneity and violations of independence can pose serious problems and result in biased parameter estimates and P-values. What's more, ecological and evolutionary data are often very messy, with a lot of noise and unequal sample sizes and missing data, which can help drive these violations. Thankfully, mixed-effects models provide us with many ways to incorporate violations of these assumptions directly into our models, allowing us to use all of our data and have greater confidence in our parameter estimates and inferences.

The RIKZ dataset

Throughout the first part of this lecture, we will be making use of the RIKZ dataset, described in Zuur *et al.* (2007) and Zuur *et al.* (2009). For each of 9 intertidal areas (denoted 'Beaches'), the researchers sampled five sites (denoted 'Sites') and at each site they measured abiotic variables and the diversity of macro-fauna (e.g. aquatic invertebrates). Here, species richness refers to the total number of species found at a given site while NAP (i.e. Normal Amsterdams Peil) refers to the height of the sampling location relative to the mean sea level and represents a measure of the amount of food available for birds, etc. For our purpose, the main question is:

1. **What is the influence of NAP on species richness?**

A diagrammatic representation of the dataset can be seen below (Modified from Zuur *et al.* (2009), Chapter 5).

Let's start by loading and examining the data.

```
library(tidyverse)
```

```
## Parsed with column specification:
## cols(
##   Richness = col_double(),
##   Exposure = col_double(),
##   NAP = col_double(),
##   Beach = col_double(),
##   Site = col_double()
## )
```

```
# Load data
rikz_data <- "https://uoftcoders.github.io/rcourse/data/rikz_data.txt"
download.file(rikz_data, "rikz_data.txt")

rikz_data <- read_delim("rikz_data.txt",
                        col_names = TRUE,
                        delim = "\t")
```

```
# Check whether Beach is encoded as factor. Convert.
is.factor(rikz_data$Beach)
```

```
## [1] FALSE
```

```
rikz_data <- rikz_data %>%
    mutate(Beach = as.factor(Beach))
```

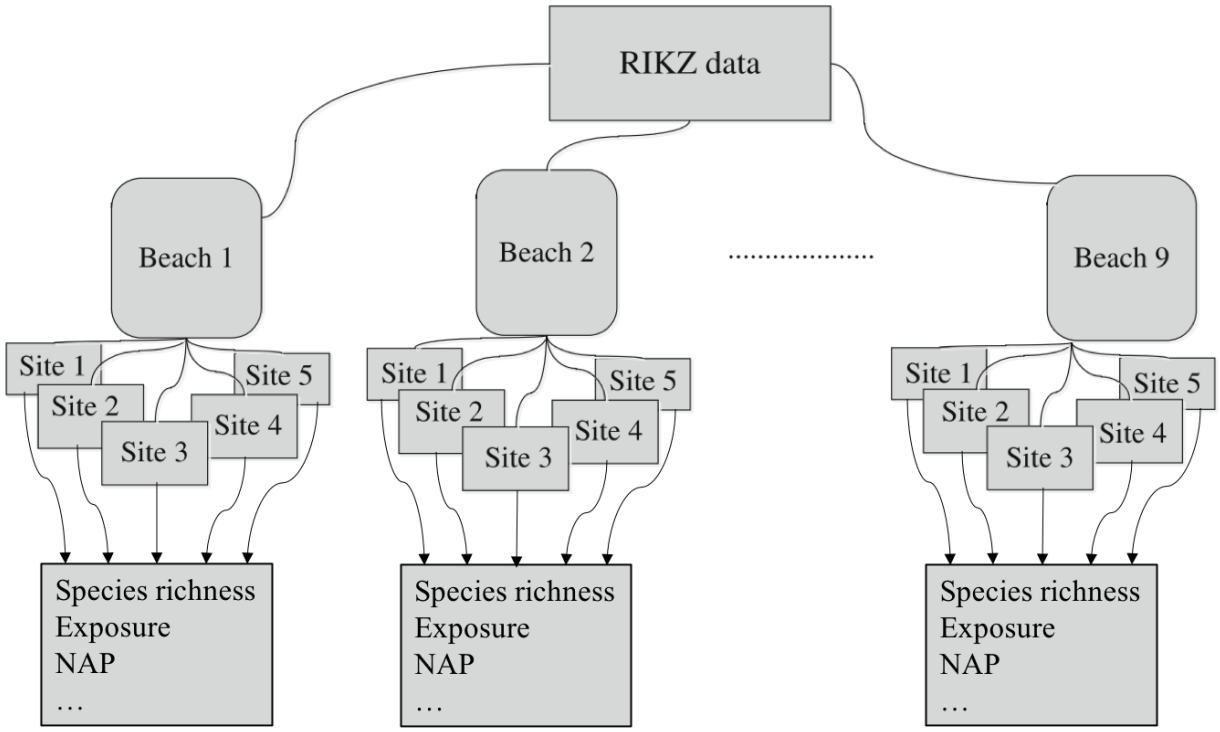


Figure 1: **Figure 1**: Diagrammatic representation of the RIKZ dataset

```
# Examine structure of the dataframe and view first 5 columns
str(rikz_data)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 45 obs. of  5 variables:
##  $ Richness: num  11 10 13 11 10 8 9 8 19 17 ...
##  $ Exposure: num  10 10 10 10 10 8 8 8 8 8 ...
##  $ NAP     : num  0.045 -1.036 -1.336 0.616 -0.684 ...
##  $ Beach   : Factor w/ 9 levels "1","2","3","4",..: 1 1 1 1 1 2 2 2 2 2 ...
##  $ Site    : num  1 2 3 4 5 1 2 3 4 5 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Richness = col_double(),
##   ..   Exposure = col_double(),
##   ..   NAP = col_double(),
##   ..   Beach = col_double(),
##   ..   Site = col_double()
##   .. )
```

```
head(rikz_data)
```

```
## # A tibble: 6 x 5
##   Richness Exposure    NAP Beach  Site
##      <dbl>    <dbl>  <dbl> <fct> <dbl>
## 1       11       10  0.045 1         1
## 2       10       10 -1.04  1         2
## 3       13       10 -1.34  1         3
```

```
## 4       11       10  0.616 1         4
## 5       10       10 -0.684 1         5
## 6        8        8  1.19  2         1
```

We can see that the data contains 45 rows (observations). As expected, these observations were taken across 9 beaches, each with 5 sites. We have encoded 'Beach' as a factor, which will facilitate plotting and its use as a random effect downstream. Let's go ahead and perform a linear regression to examine the relationship between species richness and NAP, pooling data across all beaches

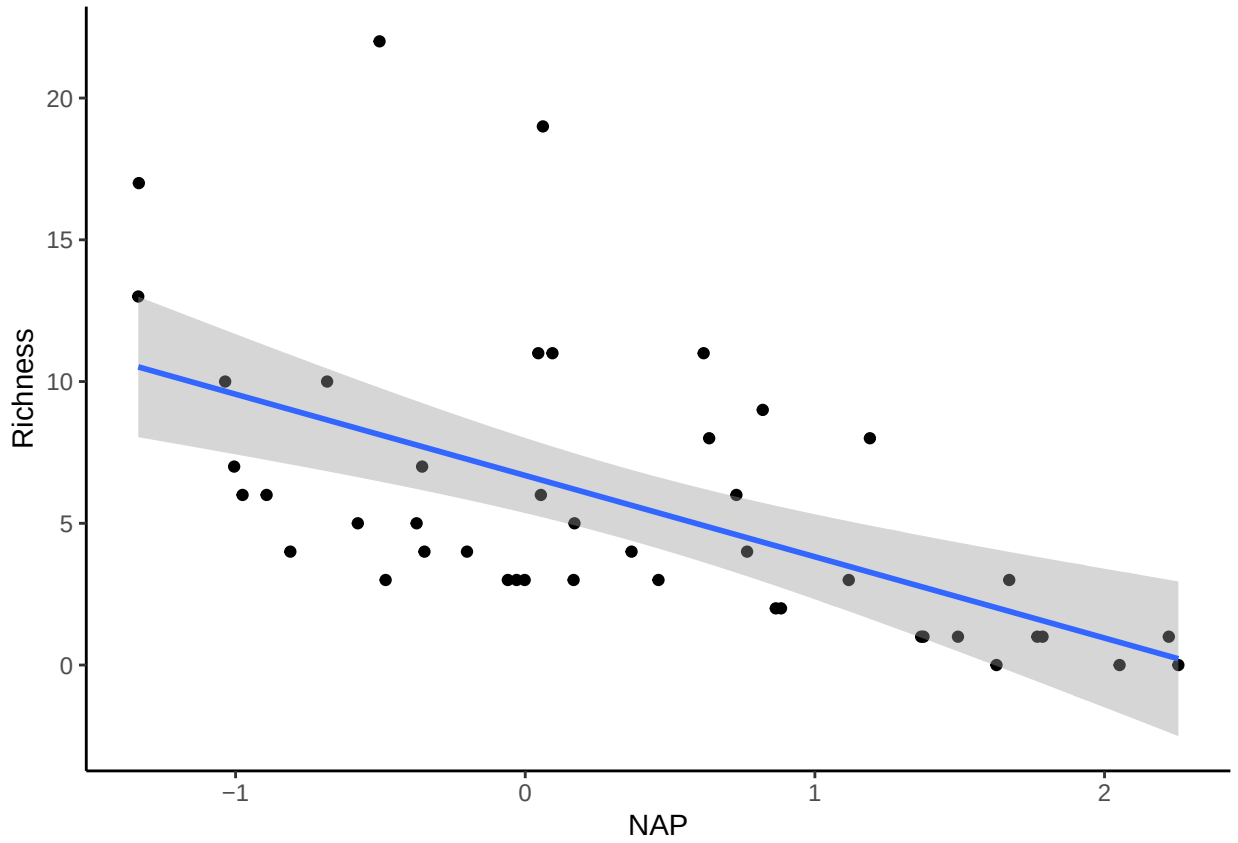
Standard linear regression

```
# Run basic linear model using all of the data.
basic.lm <- lm(Richness~ NAP, data = rikz_data)
summary(basic.lm)
```

```
##
## Call:
## lm(formula = Richness ~ NAP, data = rikz_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0675 -2.7607 -0.8029  1.3534 13.8723
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.6857     0.6578  10.164 5.25e-13 ***
## NAP          -2.8669     0.6307  -4.545 4.42e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.16 on 43 degrees of freedom
## Multiple R-squared:  0.3245, Adjusted R-squared:  0.3088
## F-statistic: 20.66 on 1 and 43 DF,  p-value: 4.418e-05
```

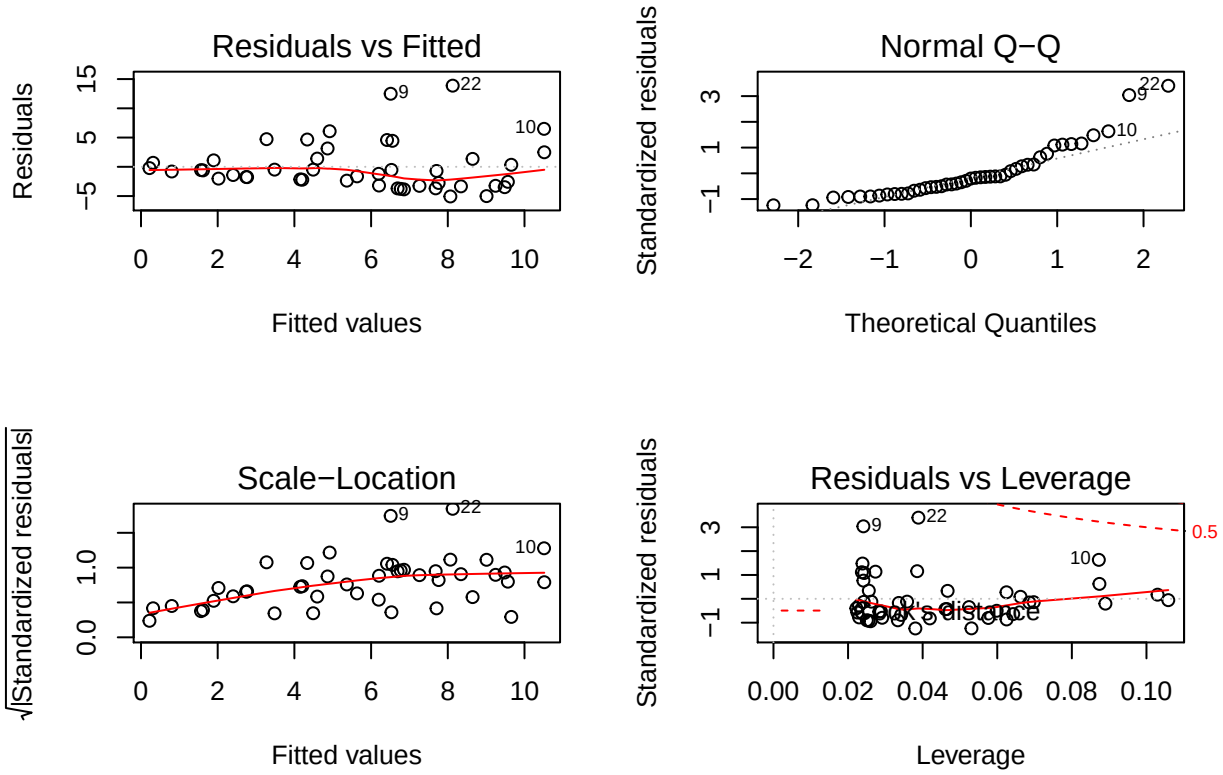
From the model output above, it looks like NAP is significantly, negatively (Estimate < 0) associated with species richness. Let's plot this relationship to see what it looks like.

```
# Plot relationship from above model
ggplot(rikz_data, aes(x = NAP, y = Richness)) +
    geom_point() +
    geom_smooth(method = "lm") +
    theme_classic()
```



However, before we trust this result, we should confirm that the assumptions of the linear regression are met. We can plot the residuals against the fitted values (homogeneity and independence) and a QQ-plot (normality). Thankfully, R's base plotting function does all of this work for us.

```
# Check assumptions.
# Normality homogeneity of variance violated
par(mfrow=c(2,2))
plot(basic.lm)
```



The first and third panels suggest that the homogeneity assumption is violated (increasing variance in the residuals with increasing fitted values). Similarly, panel 2 suggests non-normality (points falling off of the dotted line). A third-root transformation of the response variable (i.e. Richness) seems to alleviate both of these problems.

- **Tip:** Right skewed response variables can be normalized using root transformations (e.g. square root, log, third-root, etc.), with greater roots required for increasingly right-skewed data. Left skewed response variables can be normalized with power transformations (e.g. squared, 3rd power, etc.)

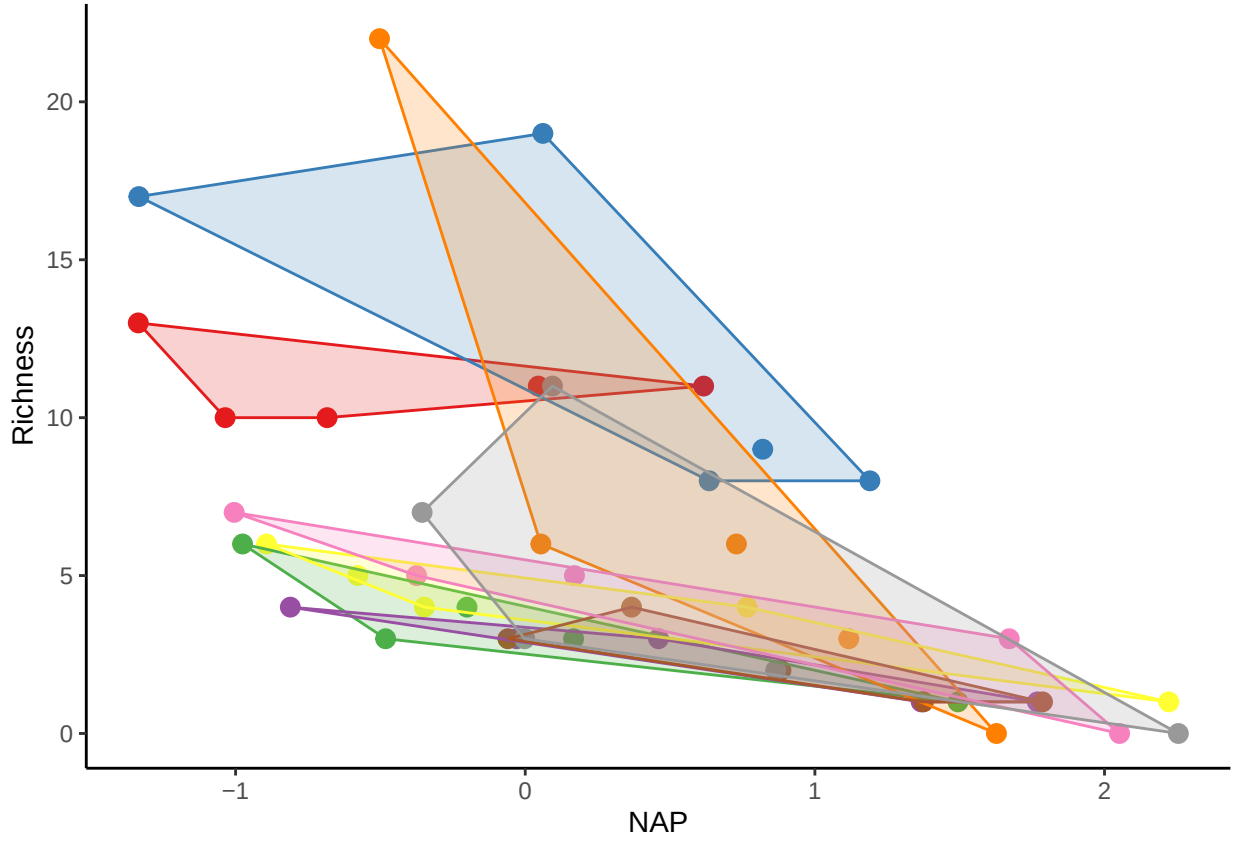
Nonetheless, for the analyses in this section, we will ignore violations of these assumptions for the purpose of better illustrating mixed-effects modelling strategies on untransformed data. In fact, these data violate another, potentially more problematic assumption; namely, the observations are not independent.

Non-independence of observations

Remember, the species richness data come from multiple sites within multiple beaches. While each beach may be independent, sites within a beach are likely to have similar species richness due simply to their proximity within the same beach. In other words, observations among sites within a beach are **not independent**. Another way of saying this is that the data are **nested**. Nesting in this sense is a product of the experimental design (i.e. we chose to sample 5 sites within each beach) and not necessarily of the data itself. Other types of nested data include: sampling the same individual pre- and post-treatment or sampling them multiple times (i.e. repeated measures), or sampling multiple tissues from the same individuals. We can visualize the non-independence of observation within the same beach by producing a figure similar to the one above but clustered by beach (each beach is a different colour).

```
library(plyr)
# Function to find polygons
```

```
find_hull <- function(df) df[chull(df$Richness, df$NAP), ]
# Identify polygons in data
hulls <- ddply(rikz_data, "Beach", find_hull)
# Plot
ggplot(rikz_data, aes(x = NAP, y = Richness, colour = Beach)) +
    geom_point(size = 3) +
    theme_classic() +
    theme(legend.position = "none") +
    scale_colour_brewer(palette="Set1") +
    scale_fill_brewer(palette="Set1") +
    geom_polygon(data=hulls, aes(fill = Beach), alpha = 0.2)
```



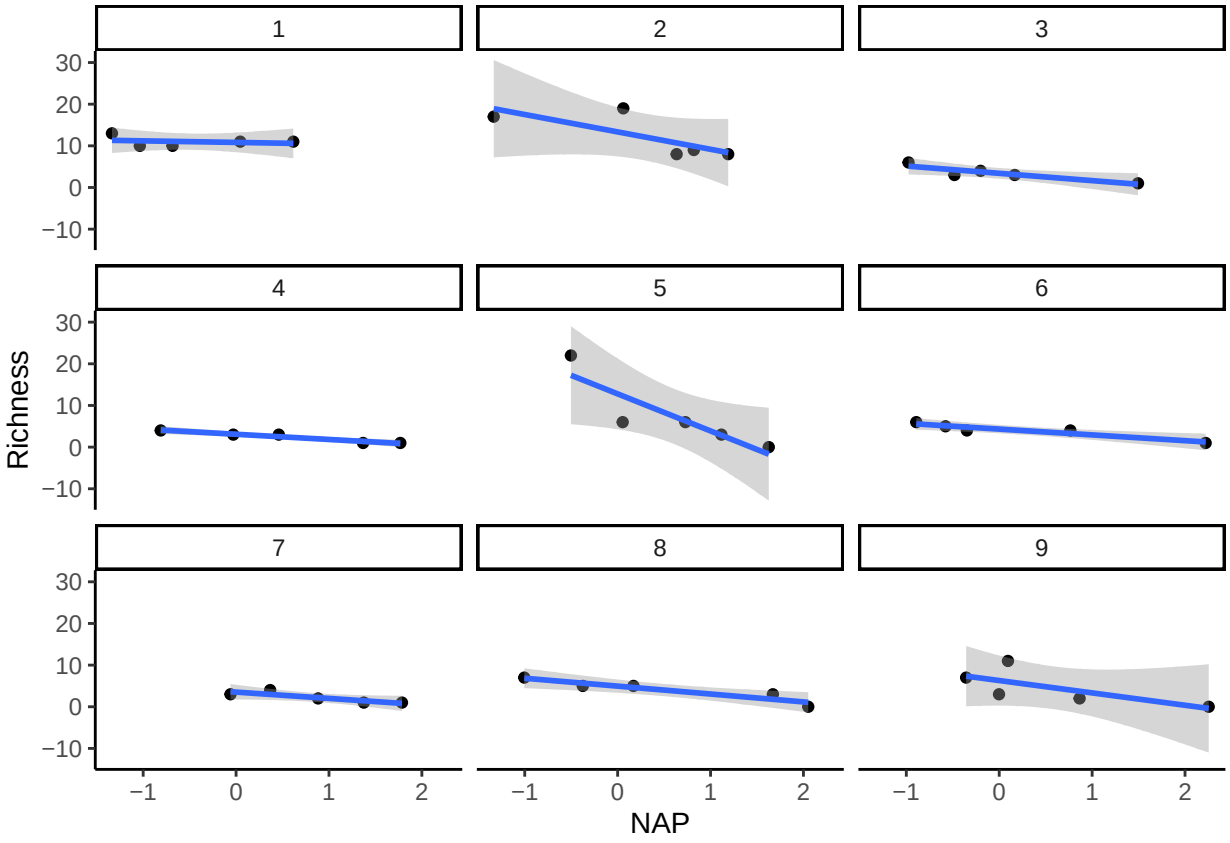
As you can see, observations from the same beach tend to cluster together. If the observations were independent, all of the clusters would overlap. Instead, observations from one beach are one average more similar to those within the same beach than to those from other beaches. We need to account for this non-independence in our modelling.

Accounting for non-independence

One approach to account for this non-independence would be to run a separate analysis for each beach, as shown in the figure and table below.

```
# We could account for non-independence by running a separate analysis
# for each beach
ggplot(rikz_data, aes(x = NAP, y = Richness)) +
    geom_point() +
```

```
    geom_smooth(method = "lm") +
    facet_wrap(~ Beach) +
    xlab("NAP") + ylab("Richness") +
    theme_classic()
```




```
# Run linear model of Richness against NAP for each beach
beach_models <-  rikz_data %>%
    group_by(Beach) %>%
    do(mod = lm(Richness ~ NAP, data = .))

# Get the coefficients by group in a tidy data_frame
library(broom)
dfBeachModels = tidy(beach_models, mod)
dfBeachModels %>%
    filter(term == "NAP") %>%
    dplyr::select(estimate, p.value)
```

```
## # A tibble: 9 x 3
## # Groups:   Beach [9]
##   Beach estimate p.value
##   <fct>    <dbl>   <dbl>
## 1 1       -0.372 0.694
## 2 2       -4.18  0.128
## 3 3       -1.76  0.0363
## 4 4       -1.25  0.00613
## 5 5       -8.90  0.0484
```

```
## 6 6       -1.39  0.0148
## 7 7       -1.52  0.0579
## 8 8       -1.89  0.0170
## 9 9       -2.97  0.185
```

However, the issue with this approach is that each analysis only has 5 points (a really low sample size!) and we have to run multiple tests. As such, we run the risk of obtaining spuriously significant results purely by chance. This is not the best approach. Perhaps instead we can simply include a term for beach in our model, thereby estimating its effects. This is done in the model below.

```
basic.lm <- lm(Richness ~ NAP + Beach, data = rikz_data)
summary(basic.lm)
```

```
##
## Call:
## lm(formula = Richness ~ NAP + Beach, data = rikz_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8518 -1.5188 -0.1376  0.7905 11.8384
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.8059     1.3895   7.057 3.22e-08 ***
## NAP          -2.4928     0.5023  -4.963 1.79e-05 ***
## Beach2        3.0781     1.9720   1.561  0.12755
## Beach3       -6.4049     1.9503  -3.284  0.00233 **
## Beach4       -6.0329     2.0033  -3.011  0.00480 **
## Beach5       -0.8983     2.0105  -0.447  0.65778
## Beach6       -5.2231     1.9682  -2.654  0.01189 *
## Beach7       -5.4367     2.0506  -2.651  0.01196 *
## Beach8       -4.5530     1.9972  -2.280  0.02883 *
## Beach9       -3.7820     2.0060  -1.885  0.06770 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.06 on 35 degrees of freedom
## Multiple R-squared:  0.7025, Adjusted R-squared:  0.626
## F-statistic: 9.183 on 9 and 35 DF,  p-value: 5.645e-07
```

That's a lot of terms! Note what's happening here. The model is estimating a separate effect for each level of beach (8 total since 1 is used as the reference). Because we need one degree of freedom to estimate each of these, that's a total of 8 degrees of freedom! In this case, it had little effect of changing our interpretation of the effects of NAP on richness (which is still negative and significant). However, sometimes the inclusion of additional terms in this way will change the estimated effect of other terms in the model and alter their interpretation. The question we need to ask ourselves here is: *Do we really care about differences between beaches?* Maybe but probably not. These beaches were a random subset of all beaches that could have been chosen but we still need to account for the non-independence of observations within beaches. This is where random-effects become useful.

Random-effect models

Random intercept model

We do not want to pay the steep price of 8 degrees of freedom to include a Beach term in our model but still want to estimate the variance among beaches and must account for the non-independence of sites within beaches. We can do this by including it as a **random effect**, while NAP remains a **fixed effect**. As such, we model a separate y-intercept (i.e. Richness at NAP = 0) for each beach and estimate the variance around this intercept. A small variance means that variances per beach are small whereas a large variance means the opposite (this will become clearer shortly). We can run mixed-effects models using the `lmer` function from the `lme4` R package and obtain parameter estimates using the `lmerTest` package. The question we are now asking is:

1. **What is the influence of NAP on species richness, while accounting for variation within beaches?**

```
library(lme4)
library(lmerTest)

# Random intercept model with NAP as fixed effect and Beach
# as random effect
mixed_model_IntOnly <- lmer(Richness ~ NAP + (1|Beach),
                            data = rikz_data, REML = FALSE)
summary(mixed_model_IntOnly)
```

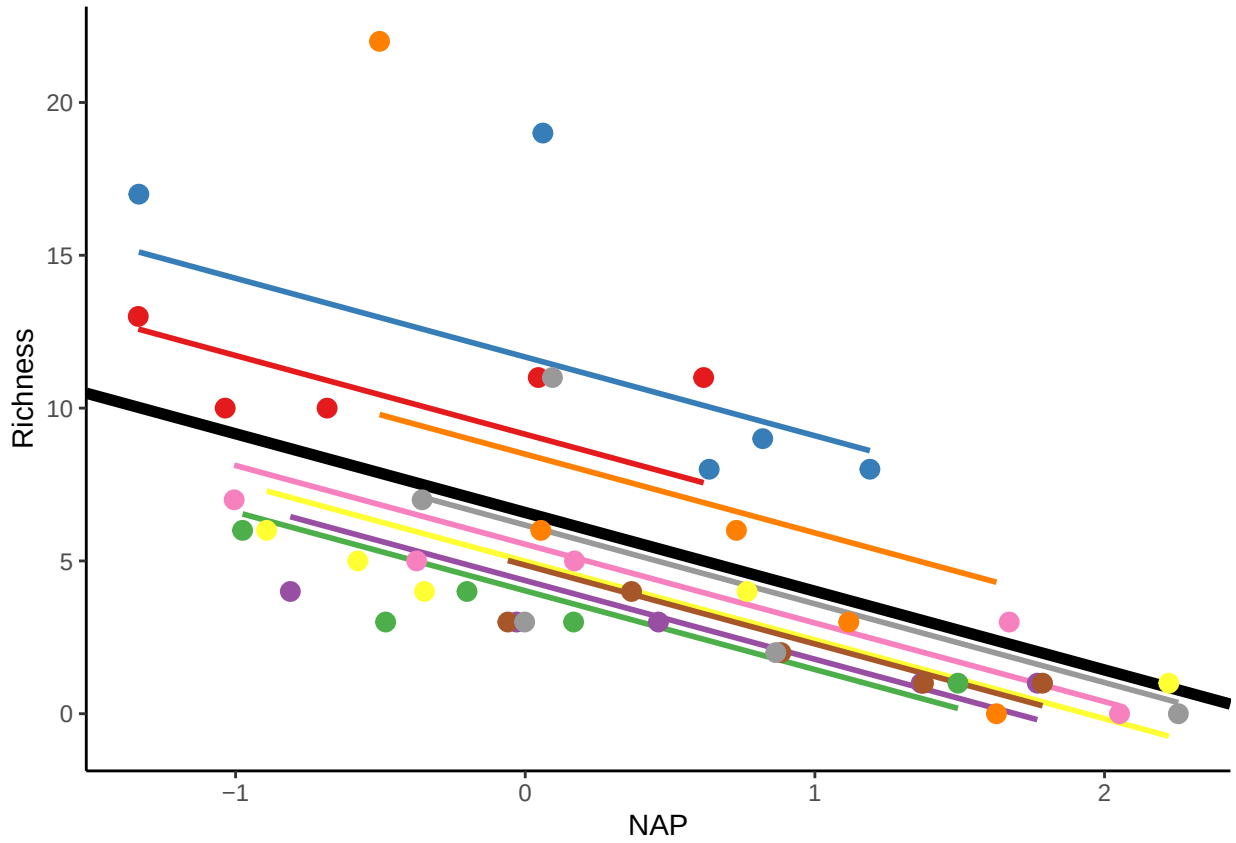
```
## Linear mixed model fit by maximum likelihood . t-tests use
##   Satterthwaite's method [lmerModLmerTest]
## Formula: Richness ~ NAP + (1 | Beach)
##    Data: rikz_data
##
##      AIC      BIC   logLik deviance df.resid
##    249.8    257.1   -120.9    241.8       41
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.4258 -0.5010 -0.1791  0.2452  4.0452
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Beach    (Intercept) 7.507    2.740
##  Residual             9.111    3.018
## Number of obs: 45, groups:  Beach, 9
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)   6.5844     1.0321  9.4303   6.380 0.000104 ***
## NAP          -2.5757     0.4873 38.2433  -5.285 5.34e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##     (Intr)
## NAP -0.164
```

The `(1|Beach)` is the random effect term, where the `1` denotes this is a random-intercept model and the term on the right of the `|` is a nominal variable (or factor) to be used as the random effect. Note that it's

considered best practice that random effects have at least 5 levels, otherwise it should be used as a fixed effect (we have 9 so we're good!). From top to bottom, the output is telling us that the model was fit using maximum likelihood with parameters estimated using the Satterthwaite approximation. It then gives us the AIC, BIC and log-likelihood values for this particular model. I'm sure most of this sounds like jargon. Don't worry! These will become clearer later and AIC will be especially useful for our lecture on Model Selection (Lecture 9). The next part is important: it gives us the estimated variance for the random effects in the model. Here, it's telling us that the variance associated with the effect of beach is `7.507`. In other words, differences between beaches account for `(7.507 / 7.507 + 9.111) * 100 = 45%` of the residual variance *after accounting for the fixed effects* in the model. Note the denomiator here is the **total variance** (i.e. the sum of the variance components from all the random effects, including the residuals). Finally, we have the fixed effect portions of the model, with a separate intercept, slope and P-value for the effect of NAP. Let's visualize the fitted values for this model.

```
# Let's predict values based on our model and add these to our dataframe
# These are the fitted values for each beach, which are modelled separately.
rikz_data$fit_InterceptOnly <- predict(mixed_model_IntOnly)

# Let's plot the
ggplot(rikz_data, aes(x = NAP, y = Richness, colour = Beach)) +
    # Add fixed effect regression line (i.e. NAP)
    geom_abline(aes(intercept = `(Intercept)`, slope = NAP),
                size = 2,
                as.data.frame(t(fixef(mixed_model_IntOnly)))) +
    # Add fitted values (i.e. regression) for each beach
    geom_line(aes(y = fit_InterceptOnly), size = 1) +
    geom_point(size = 3) +
    theme_classic() +
    theme(legend.position = "none") +
    scale_colour_brewer(palette="Set1")
```



The thick black line corresponds to the fitted values associated with the fixed-effect component of the model (i.e. `6.58 x -2.58(x)`). The thin coloured lines correspond to the fitted values estimated for each beach. As you can see, they all have separate intercepts, as expected. As the estimated variance of the random effect increases, these lines would become more spread around the thick black line. If the variance was 0, all the coloured lines would coincide with the thick black line.

Random intercept-slope model

The model above allows the intercept for each beach to vary around the population-level intercept. However, what if beaches don't only vary in their mean richness, but the richness on each beach also varies in its response to NAP. In standard regression terms, this would amount to including NAP, Beach and NAP x Beach effects in the model. Of course, including such fixed-effects here would consume way too many degrees of freedom and we already decided we don't really care about differences between beaches. Thankfully, we can still allow beaches to vary in the response to NAP using a **random intercept-slope model**. We can fit the random intercept-slope model to these data using the code below.

```
# Random intercept and slope model
mixed_model_IntSlope <- lmer(Richness ~ NAP + (1 + NAP|Beach),
                             data = rikz_data, REML = FALSE)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
## $checkConv, : Model failed to converge with max|grad| = 0.00629499 (tol =
## 0.002, component 1)
```

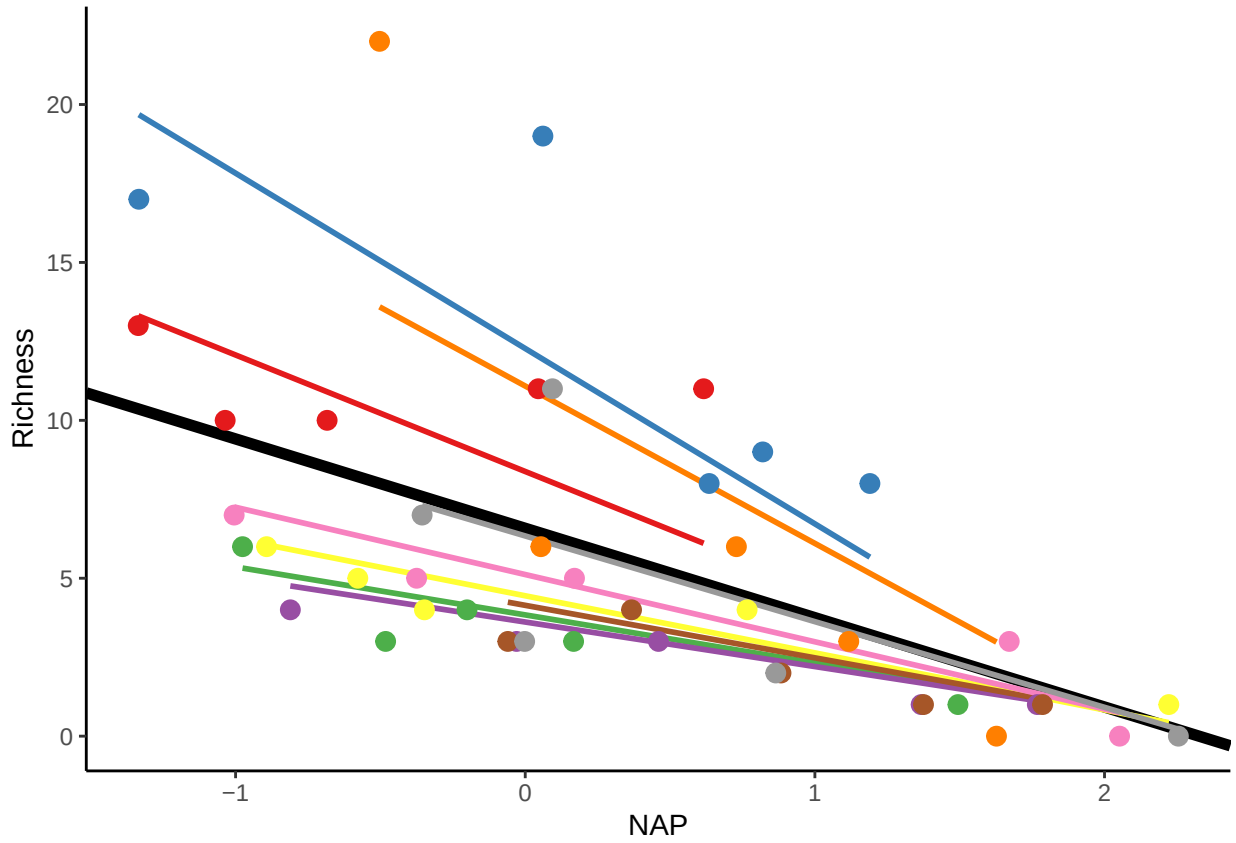
```
summary(mixed_model_IntSlope)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use
```

```
##   Satterthwaite's method [lmerModLmerTest]
## Formula: Richness ~ NAP + (1 + NAP | Beach)
##    Data: rikz_data
##
##      AIC      BIC   logLik deviance df.resid
##    246.7    257.5   -117.3    234.7       39
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.7985 -0.3419 -0.1827  0.1747  3.1386
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Beach    (Intercept) 10.944   3.308
##           NAP          2.502   1.582    -1.00
##  Residual              7.175   2.679
## Number of obs: 45, groups:  Beach, 9
##
## Fixed effects:
##             Estimate Std. Error     df t value Pr(>|t|)
## (Intercept)    6.582      1.188  8.898   5.540 0.000376 ***
## NAP           -2.829      0.685  7.923  -4.131 0.003365 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##     (Intr)
## NAP -0.810
## convergence code: 0
## Model failed to converge with max|grad| = 0.00629499 (tol = 0.002, component 1)
```

The above model now allows both the intercept and slope of the relationship between Richness and NAP to vary across beaches. The only difference here is the additional variance component in the random effects, which estimates the variance in slopes across beaches. It also includes a `Cor` term, which estimates the correlation between the intercept and slope variances. At -1, this implies that beaches with larger intercepts also have more steeply negative slopes, as can be seen in the figure below.

```
rikz_data$fit_IntSlope <- predict(mixed_model_IntSlope)
ggplot(rikz_data, aes(x = NAP, y = Richness, colour = Beach)) +
    geom_abline(aes(intercept = `(Intercept)`, slope = NAP),
                size = 2,
                as.data.frame(t(fixef(mixed_model_IntSlope)))) +
    geom_line(aes(y = fit_IntSlope), size = 1) +
    geom_point(size = 3) +
    theme_classic() +
    theme(legend.position = "none") +
    scale_colour_brewer(palette="Set1")
```



Random effects only model

Note that it is not always necessary to specify fixed effects, in the same way that it is not always necessary to specify random effects. For example, we could run the following model.

```
mixed_model_NoFix <- lmer(Richness ~ 1 + (1|Beach),
                          data = rikz_data, REML = TRUE)
summary(mixed_model_NoFix)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Richness ~ 1 + (1 | Beach)
##    Data: rikz_data
##
## REML criterion at convergence: 261.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.7797 -0.5070 -0.0980  0.2547  3.8063
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Beach    (Intercept) 10.48    3.237
##  Residual             15.51    3.938
## Number of obs: 45, groups:  Beach, 9
##
```

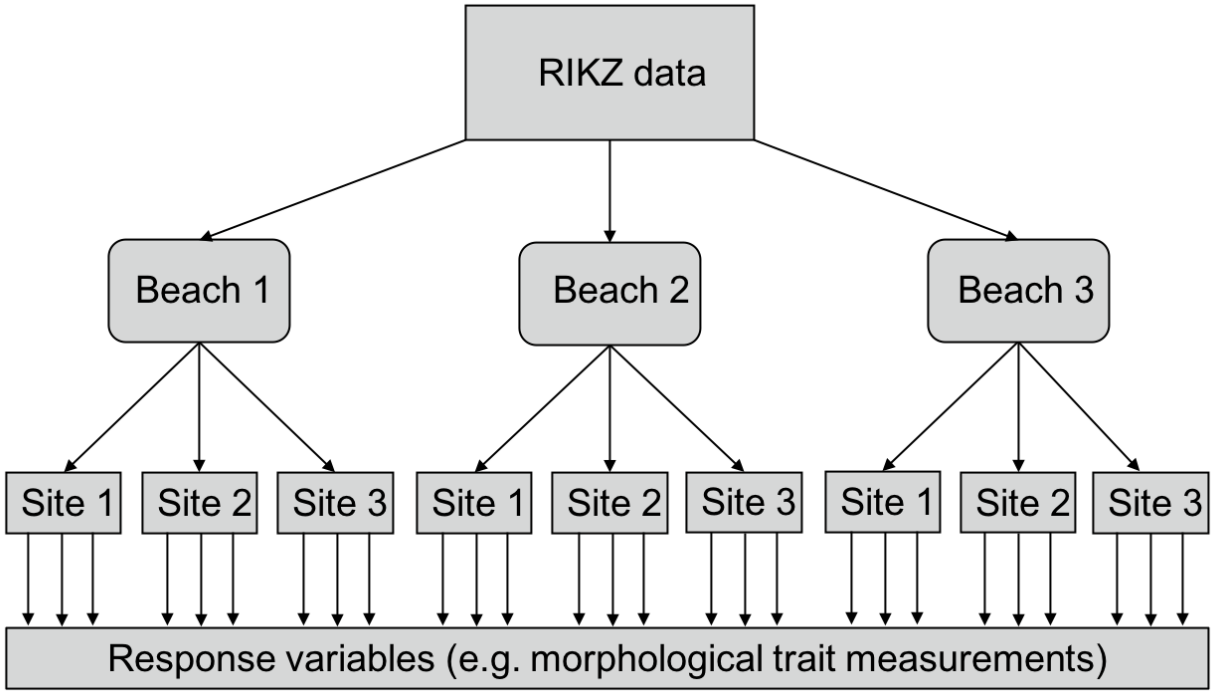


Figure 2: **Figure 2:** Diagrammatic representation of what the RIKZ data would look like if it were more deeply nested (i.e. if each site had multiple samples taken.)

```
## Fixed effects:
##             Estimate Std. Error    df t value Pr(>|t|)
## (Intercept)    5.689      1.228 8.000   4.631  0.00169 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now that we have specified these three different models, how do we know which to choose? After all, they all provide slightly different estimates for the effects of NAP (if NAP is even included) and P-values. We will come back to this question in Lecture 9 when discussing model selection.

Deeply nested and crossed effects

Have a look back at the original diagram showing the layout of the RIKZ data and the dataset itself. Every site within each beach is associated with only one observation for each of the variables (e.g. Species richness). As such, we used mixed-effects modelling to account for the variance among these 5 observations within each of the five beaches. But what about if each of those sites additionally included multiple samples (e.g. measurements of morphological traits of multiple individuals of a species), as in the diagram below?. We would the need to account for the variance both within sites and within beaches.

Thankfully, `lmer` allows us to do this quite easily. To account for variation within sites **and** within beaches, we would need to modify our existing random effect term. We would write this as `(1|Beach/Site)`, which means "Site nested within beach". This expands to — and can also be written as — `(1|Beach) + (1|Beach:Site)`. Thus, we are modelling a separate intercept for every beach and for every site within beach. I should pause for a moment here and say that the 'Site 1' from 'Beach 1' is **not** the same as 'Site 1' from 'Beach 2' and this was true of the original data as well. These sites are distinct; despite carrying the same label, they are occurring on distinct beaches and are thus not the same site. **This is what makes the data nested.**

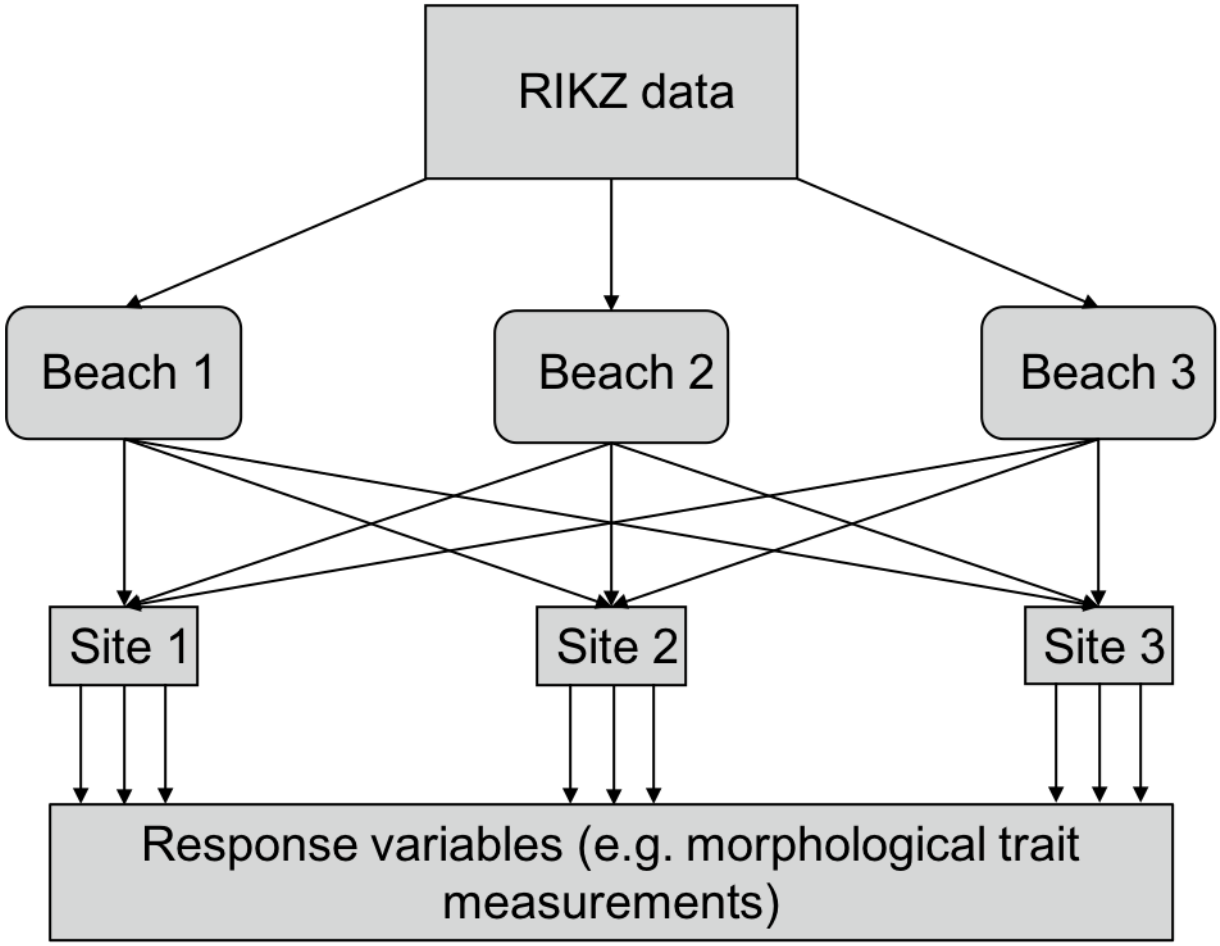


Figure 3: **Figure 3:** Diagrammatic representation of what the RIKZ dataset would look like if it were fully crossed (i.e. if every site occurred on every beach)

An alternative design to this would be **crossed effects**, in which any site could be observed on any beach. If all sites were observed on all beaches, this would be a **fully crossed effect** whereas if some sites were observed on only some beaches, this would be a **partially crossed effect**. Crossed effects are shown in the diagram below. This may sound confusing. The easiest way to think about it is that if the data are not nested, they are crossed.

I will illustrate both deeply nested and crossed random effects in a single model using data from Fitzpatrick *et al.* (2019). The authors were interested in understanding whether soil microbes can help plants cope with stress imposed by drought and how this can influence plant evolution. To do this, Fitzpatrick *et al.* grew 5 plants from each of 44 *Arabidopsis thaliana* accessions in each combination of the following treatments:

1. Well-watered and live soil (i.e. with microbes)
2. Well watered and sterile soil
3. Drought and live soil
4. Drought and sterile soil

The experiment took place inside of growth chambers at the University of Toronto, where plants were randomly assigned into trays, which were then randomly assigned onto racks in the growth chamber. Thus,

plants only occur within one tray and each tray only occurs within one rack (tray is nested within rack). Given that plants within a tray could be more similar to one another than plants from another tray (i.e. tray effect) and the same is true for racks, this non-independence needs to be accounted for. However, every accession occurred on every rack (i.e. crossed effect) and individual plants from an accession are likely to be more similar to other plants from that accession than to plants from another accession. Again, this non-independence needs to be accounted for. The authors tracked flowering date and counted the number of fruit produced by each plant (n = 879) at the end of the experiment. The code below provides a useful way of examining the data to assess whether terms are nested or crossed and then fits a mixed-effects model to the Fitzpatrick *et al.* data. The question we are interested in here is: **Do soil microbes, drought, or their interaction influence the number of fruit produced by *A. thaliana* plants?**

```
## Parsed with column specification:
## cols(
##   accession = col_double(),
##   water = col_character(),
##   microbe = col_character(),
##   replicate = col_double(),
##   tray = col_double(),
##   shelf = col_character(),
##   rack = col_double(),
##   fruit = col_double(),
##   flowering = col_double(),
##   counter = col_character()
## )
```

```
# Load in and examine data
Fitz_data <- "https://uoftcoders.github.io/rcourse/data/Fitzpatrick_2018.csv"
download.file(Fitz_data, "Fitzpatrick_2018.csv")
Fitz_data <- read_csv("Fitzpatrick_2018.csv", col_names = TRUE)
```

```
# Let's change some columns to factors and remove rows with no fruit data
Fitz_data <- Fitz_data %>%
    mutate(
        tray = as.factor(tray),
        rack = as.factor(rack),
        accession = as.factor(accession)
    ) %>%
    filter(!is.na(fruit))

glimpse(Fitz_data)
```

```
## Observations: 879
## Variables: 10
## $ accession <fct> 10, 72, 52, 69, 56, 68, 30, 44, 49, 3, 70, 58, 33, 3...
## $ water     <chr> "D", "D", "W", "D", "W", "W", "W", "D", "D", "D", "D...
## $ microbe   <chr> "S", "L", "S", "S", "L", "S", "L", "S", "L", "S", "L...
## $ replicate <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ tray      <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2...
## $ shelf     <chr> "A", "A", "A", "A", "A", "A", "A", "A", "A", "A", "A...
## $ rack      <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ fruit     <dbl> 6, 6, 21, 37, 12, 23, 29, 5, 17, 17, 0, 0, 20, 28, 0...
## $ flowering <dbl> 31, 30, 30, 10, 7, 15, 15, 13, 6, 19, NA, NA, 10, 8,...
## $ counter   <chr> "JV", "JV", "JV", "JV", "JV", "JV", "JV", "JV", "JV"...
```

```
head(Fitz_data)
```

```
## # A tibble: 6 x 10
##   accession water microbe replicate tray  shelf rack  fruit flowering
##   <fct>     <chr> <chr>       <dbl> <fct> <chr> <fct> <dbl>     <dbl>
## 1 10        D     S               1 1     A     1         6        31
## 2 72        D     L               1 1     A     1         6        30
## 3 52        W     S               1 1     A     1        21        30
## 4 69        D     S               1 1     A     1        37        10
## 5 56        W     L               1 1     A     1        12         7
## 6 68        W     S               1 1     A     1        23        15
## # ... with 1 more variable: counter <chr>
```

```
#Let's look at how trays breakdown across the levels of rack
xtabs(~ rack + tray, Fitz_data)
```

```
##     tray
## rack  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
##    1 15 15 15 15 15 15 15 15 15 15 15 11  0  0  0  0  0  0  0  0  0  0  0
##    2  0  0  0  0  0  0  0  0  0  0  0  0 15 15 15 15 15 15 15 15 15 15 15
##    3  0  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
##    5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##     tray
## rack 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
##    1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##    2 11  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##    3  0 15 15 15 15 15 15 15 15 15 15 15 11  0  0  0  0  0  0  0  0  0  0
##    4  0  0  0  0  0  0  0  0  0  0  0  0  0 15 15 15 15 15 15 15 15 15 15
##    5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##     tray
## rack 47 48 49 50 51 52 53 54 55 56 57 58 59 60
##    1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##    2  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  0  0
##    4 15 11  0  0  0  0  0  0  0  0  0  0  0  0
##    5  0  0 14 15 15 15 15 15 15 15 15 15 15 11
```

As you can see above, each tray occurs in only one rack. This is an indication that the data are nested (i.e. tray is nested within racks). Let's now look at how accessions breakdown across the racks

```
# Breakdown of accessions across the racks
xtabs(~ rack + accession, Fitz_data)
```

```
##     accession
## rack 1 3 8 9 10 11 12 15 19 21 22 30 32 33 37 38 41 42 43 44 45 47 49 50
##    1 4 4 4 4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4
##    2 4 4 4 4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4
##    3 4 4 4 4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4
##    4 4 4 4 4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4
##    5 4 4 4 4  4  4  3  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4
##     accession
## rack 51 52 54 55 56 58 60 61 62 63 64 65 66 67 68 69 70 71 72 73
##    1  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4
##    2  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4
##    3  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4
```

```
##    4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4
##    5  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4
```

In this case, we see that every accession occurs on every rack. In other words, accession and rack are fully crossed. Let's go ahead and fit our model.

```
# Fit mixed-effects model
Fitz_mod <- lmer(fruit ~ water + microbe + water:microbe +
                   (1|rack/tray) + (1|rack:accession),
                 data = Fitz_data)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
## $checkConv, : Model failed to converge with max|grad| = 0.00267274 (tol =
## 0.002, component 1)
```

```
summary(Fitz_mod)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: fruit ~ water + microbe + water:microbe + (1 | rack/tray) + (1 |
##     rack:accession)
##    Data: Fitz_data
##
## REML criterion at convergence: 6983.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.4963 -0.5763 -0.0542  0.4587  6.5849
##
## Random effects:
##  Groups          Name        Variance Std.Dev.
##  rack:accession (Intercept)  72.524   8.516
##  tray:rack      (Intercept)   9.493   3.081
##  rack           (Intercept)   1.270   1.127
##  Residual                   116.806  10.808
## Number of obs: 879, groups:  rack:accession, 220; tray:rack, 60; rack, 5
##
## Fixed effects:
##                 Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)      13.2742     1.1374   8.7212  11.670 1.28e-06 ***
## waterW            6.3453     1.0568 652.9691   6.004 3.19e-09 ***
## microbeS         -0.7679     1.0558 648.7894  -0.727  0.46727
## waterW:microbeS   4.1515     1.5007 655.1532   2.766  0.00583 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) waterW micrbS
## waterW      -0.465
## microbeS    -0.463  0.500
## watrW:mcrbS  0.327 -0.705 -0.708
## convergence code: 0
## Model failed to converge with max|grad| = 0.00267274 (tol = 0.002, component 1)
```

Notice we are now estimating 3 random-effect variance components. First, we are estimating the variance among accessions, which explains the most residual variance among the random effects (`(72.510 / 200.1)`

`* 100 = 58%`). We then estimate the variance between trays, followed by the variance between racks (which has only a minor effect). From the fixed effects, we see that the watering treatment had a significant effect on the number of fruit produced (main effect of `water`), although its effect depended on the presence of microbes (`water:microbe` interaction). While we will not go into the details of these results here, this example serves to illustrate how nested and crossed random effects are encoded in linear-mixed effects models.

Additional considerations

Fixed vs. random effects

- Fixed effects are the things you care about and want to estimate. You likely chose the factor levels for a specific reason or measured the variable because you are interested in the relationship it has to your response variable.
- Random effects can be variables that were opportunistically measured whose variation needs to be accounted for but that you are not necessarily interested in (e.g. spatial block in a large experiment). The levels of the random effect are likely a random subset of all possible levels (although there should be at least 5). However, if the experimental design includes nesting or non-independence of any kind, this needs to be accounted for to avoid pseudoreplication.

REML or ML

The math behind maximum likelihood (ML) and restricted maximum likelihood (REML) is beyond this course. Suffice it to say that if your interested in accurately estimating the random effects, you should fit the model with REML whereas if you're interested in estimating the fixed effects, you should fit the model with ML. Given that we are often most interested in the fixed effects, ML is usually the most appropriate (but see Lecture 9: model selection). The difference in the estimates obtained by ML and REML increases as the number of parameters in the model increases.

Other models

Mixed-effects models are very powerful when correctly applied. For example, they allow modelling of non-linear relationships (e.g. GAM models), modelling separate variances across factor levels to account for heterogeneity of variance, fitting alternative error structures to account for temporal and spatial non-independence, etc. This lecture covered the type of mixed-effects model that is most often encountered by ecologists and evolutionary biologists (at least in my experience). Nonetheless, if you find yourself needing something more, I'll refer you to the reading list below.

Additional reading

1. Zuur, A. *et al.* 2009. Mixed effects models and extensions in ecology with R. *Springer*
2. Bates, D. *et al.* 2015. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software* 67: 1-48.
3. Fitzpatrick, C. R., Mustafa, Z., and Viliunas, J. Soil microbes alter plant fitness under competition and drought. *Journal of Evolutionary Biology* 32: 438-450.
4. Crossed vs. Nested random effects
5. Fixed vs. Random effects
6. ML vs. REML