Mixed-Effects models in R

Remko Duursma[1], *Jeff Powell*[1]
[1]Hawkesbury Institute for the Environment
University of Western Sydney

WESTERN SYDNEY
UNIVERSITY

Hawkesbury Institute
for the Environment

April 14, 2016

Preface

This book is a collection of advanced topics in R. References to previous sections, as well as descriptions of datasets, are to the book by Duursma, Powell and Stone: *Data analysis and visualization with R*.

Contents

Chapter 1

Mixed-effects models

In Chapter 7, we used linear models to estimate ’fixed’ effects, which consist of specific and repeatable categories/variables that are representative of an entire population (e.g., species, age). In longitudinal studies (repeated measures) and in studies using hierarchical (nested) sampling, it is also possible to estimate effects associated with individuals sampled at random from the population of interest. These are ’random’ effects and convey information about the degree that individuals in a population differ but not how or why they differ.

One way to differentiate fixed and random effects is that fixed effects contain levels that are informative beyond the current analysis (e.g., a species of tree or a specific management type) while random effects contain levels that are not informative beyond the current analysis (e.g., a group of trees within an observation plot or a field under specific management). Another way to understand the difference is that fixed effects influence the mean of the response while random effects influence the variance of the response.

Mixed-effects models estimate both fixed and random effects and are particularly useful when dealing with potential pseudoreplication and unbalanced designs. Including random effects can also account for variation that could mask patterns in an analysis considering only fixed effects.

Two commonly used packages for fitting mixed-effects models are `nlme` and `lme4`. In this chapter, we will use the newer `lme4` package, but note that more complex correlation structures are only possible with the `nlme` package.

Rather than to present the theory underlying mixed-effects models, which is very complex, we will treat this topic by example, and thus aim at a practical application.

1.0.1 A note about p-values

The `lme4` package does not report p-values. The developers made this decision because p-values require calculating degrees of freedom. Random effects don’t necessarily have to expend the same degrees of freedom as treating them as fixed effects, so the package developers have decided not to fudge this by calculating them for you.

You do have options though. See the help page `?pvalues` for references to several options to calculate p-values with `lme4`.

In this chapter, we use two approaches to calculate p-values of fixed effects. To obtain the p-values of all fixed effects in a model, we use the `Anova` function from the `car` package (note that this function is distinct from `anova`!), but note that there are alternatives (and even when using `Anova`, you have alternatives).

The second approach is to use likelihood-ratio tests to test the significance of a single fixed effect in a model. To do this, we fit two models (one with, and one without the fixed effect of interest), and use `anova` on the two models to calculate a p-value for the fixed effect.

We recommend you try multiple methods to make sure that the main conclusions you draw from the results are robust.

1.1 A simple equation for a mixed-effects model

Before we fit a mixed-effects model, let's write down an equation that illustrates the roles of the fixed effects and random effects in the model.

For a simple linear regression of Y versus X (where X is numeric), we fit the model:

$$Y = \beta_0 + \beta_1 * X$$

where the β's are the intercept and the slope of the fitted regression line.

A mixed-effects model additionally fits two random parameters, and we can write the model as:

$$Y = (\beta_0 + b_0) + (\beta_1 + b_1) * X$$

where b_0 is the random intercept (which is normally distributed with mean zero, and some standard deviation), and b_1 is the random slope (also assumed to be normally distributed with mean of zero and some standard deviation).

When we fit a mixed-effects model, not only do we get an estimate of the variance (or standard deviation) of the random effects, we also get estimates of this random effect for each individual. These estimates are known as the BLUPs (best linear unbiased predictors).

1.2 Example: individual-level variation

The following example uses the `pref` data (not yet described in Appendix A, download the file 'prefdata.csv'). The dataset contains measurements of leaf mass per area (LMA), and distance from the top of the tree (dfromtop) on 35 trees of two species. We want to know whether LMA decreases with `dfromtop`, as expected, and whether this decrease in LMA with distance from top differs by species.

```
# Read the data and inspect the first few rows, and the species factor variable.
pref <- read.csv("prefdata.csv")
head(pref)
```

```
##     ID         species dfromtop totheight height      LMA    narea
## 1 FP11 Pinus ponderosa     8.88     22.40  13.52 319.4472 2.779190
## 2 FP11 Pinus ponderosa     0.62     22.40  21.78 342.7948 4.010700
## 3 FP11 Pinus ponderosa     4.72     22.40  17.68 329.5399 3.365579
## 4 FP15 Pinus ponderosa     2.74     27.69  24.95 312.4467 3.682907
## 5 FP15 Pinus ponderosa     5.48     27.69  22.21 278.4037 2.524224
## 6 FP15 Pinus ponderosa     8.40     27.69  19.29 255.9716 2.351546
```

```
levels(pref$species)
```

```
## [1] "Pinus monticola" "Pinus ponderosa"
```

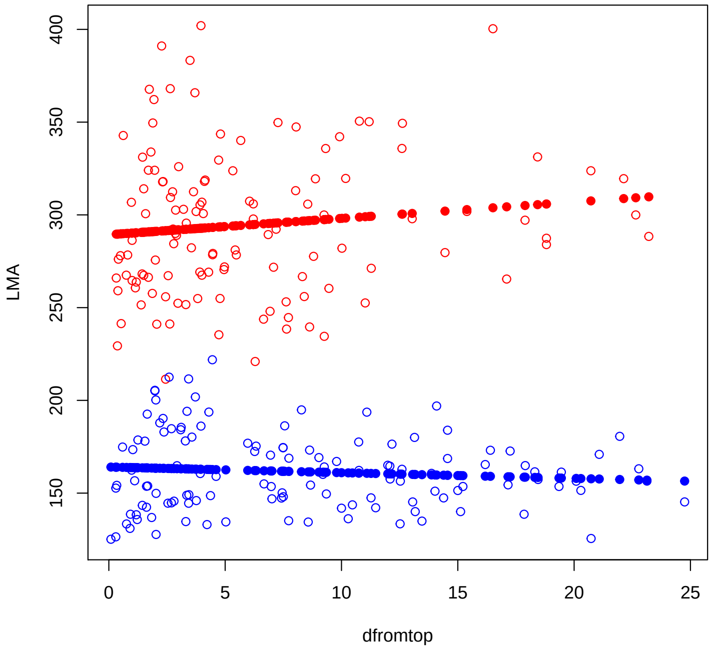

Figure 1.1: Leaf mass per area as a function of tree species (two colours) and the distance from the top of each tree. The filled symbols represent the values predicted from the model, and the open symbols the data.

Before we fit mixed-effects models, let's start with a linear regression that includes `dfromtop` and `species` as the predictor variables to observe the general patterns. We will use `predict.lm` to get the predicted values from the fitted model, and visualize how `LMA` and `dfromtop` vary from tree to tree. The following code producesFig. 1.1.

```r
# Fit a linear regression by species (ignoring individual-level variation)
lm1 <- lm(LMA ~ species + dfromtop + species:dfromtop, data=pref)

# Predicted (fitted) LMA
pref$LMApred1 <- predict(lm1, pref)

# Plot of the predictions for the two species.
palette(c("blue","red"))
with(pref, plot(dfromtop, LMA, pch=1, col=species))
with(pref, points(dfromtop, LMApred1, pch=19, col=species))
```

As we can see in Fig. 1.1, there is a strong effect of species, but it appears that `LMA` and `dfromtop` are not significantly correlated.

Try this yourself Look at the `anova` table for the fitted linear regression model from the example above to confirm that LMA does not significantly change with dfromtop.

To see whether the relationship between `LMA` and `dfromtop` potentially varies from tree to tree, we fit a linear regression separately for each tree using the `lmList` function in the `lme4` package and then plot the outcome (in Fig. 1.2).

```
# For the lmList function (Note: the nlme package also includes the lmList function)
library(lme4)

# fit linear regression by tree ('ID')
lmlis1 <- lmList(LMA ~ dfromtop | ID, data=pref)

# Extract coefficients (intercepts and slopes) for each tree
liscoef <- coef(lmlis1)

# load plottix for the 'ablineclip' function, which clips lines within the range of x
library(plotrix)

# split pref by tree (prefsp is a list)
prefsp <- split(pref, pref$ID)

# Plot
with(pref, plot(dfromtop, LMA, col=species))
for(i in 1:length(prefsp)){

  # Find min and max values of dfromtop, to send to ablineclip
  xmin <- min(prefsp[[i]]$dfromtop)
  xmax <- max(prefsp[[i]]$dfromtop)

  # add regression lines
  ablineclip(liscoef[i,1], liscoef[i,2], x1=xmin, x2=xmax)
}
```

From the figure we can conclude (informally) that:

1. Intercepts vary a lot between trees
2. There seems to be a negative relationship for many trees
3. It seems there is less variation between slopes than intercepts

We know the individual data points are not independent, as they are nested within trees (that is, multiple samples were collected for each tree). To properly account for this non-independence, we have to use a mixed-effects model. In the example below, we will fit two models: one with a random intercept only, and one with a random intercept and slope.

To specify random effects with `lmer`, we add it to the formula in the right-hand side. For example, a random intercept for 'ID' (that is, the intercept will vary randomly among ID's) is coded as `(1|ID)`. If we also allow the slope of the relationship to vary, we specify it as `(dfromtop|ID)` so that the slope and intercept of the relationship between LMA and dfromtop will vary randomly between tree ID's.

```
# Random intercept only
lme1 <- lmer(LMA ~ species + dfromtop + species:dfromtop + (1|ID), data=pref)

# Random intercept and slope
lme2 <- lmer(LMA ~ species + dfromtop + species:dfromtop + (dfromtop|ID), data=pref)

# The AIC and a likelihood-ratio test tell us that we don't need a random slope.
# lower AIC indicates that model fit is better (more efficient)
AIC(lme1, lme2)

##      df      AIC
## lme1  6 2251.997
```

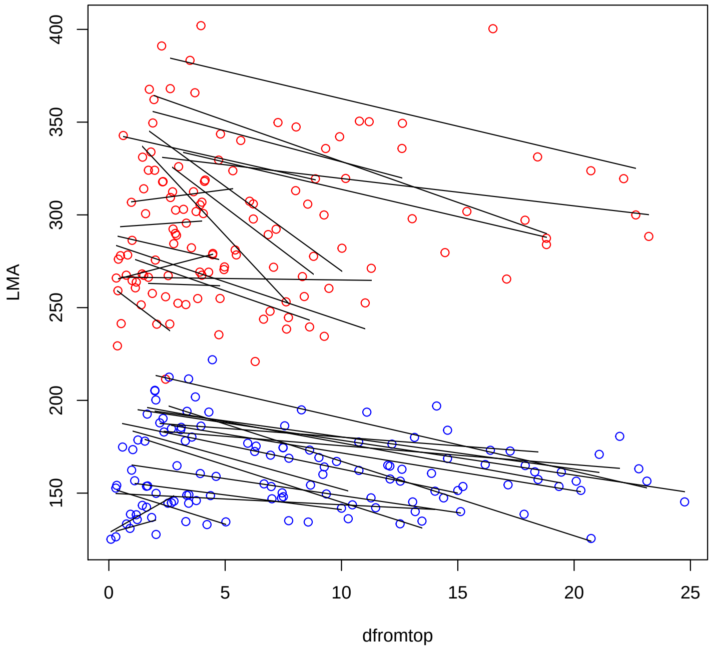

Figure 1.2: Leaf mass per area as a function of tree species (two colours) and the distance from the top of each tree. The solid lines represent the slope of the relationship for each individual tree.

```
## lme2  8 2255.735

# Likelihood ratio test : the more complex model is not supported by the data.
# Note: the models will be re-fitted with ML instead of REML; this is necessary
# when performing likelihood-ratio tests.
anova(lme1, lme2)

## Data: pref
## Models:
## lme1: LMA ~ species + dfromtop + species:dfromtop + (1 | ID)
## lme2: LMA ~ species + dfromtop + species:dfromtop + (dfromtop | ID)
##      Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lme1  6 2263.2 2284.3 -1125.6   2251.2
## lme2  8 2267.1 2295.2 -1125.5   2251.1 0.1274      2     0.9383

# Output from the random intercept model
summary(lme1)

## Linear mixed model fit by REML ['lmerMod']
## Formula: LMA ~ species + dfromtop + species:dfromtop + (1 | ID)
##    Data: pref
##
## REML criterion at convergence: 2240
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.3476 -0.5146 -0.0901  0.4656  3.3215
```

```
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  ID       (Intercept) 943.4    30.71
##  Residual             327.2    18.09
## Number of obs: 249, groups:  ID, 35
##
## Fixed effects:
##                                  Estimate Std. Error t value
## (Intercept)                      177.9437     8.0193  22.189
## speciesPinus ponderosa           134.9772    11.2094  12.041
## dfromtop                          -2.0001     0.3030  -6.602
## speciesPinus ponderosa:dfromtop  -0.8197     0.5152  -1.591
##
## Correlation of Fixed Effects:
##             (Intr) spcsPp dfrmtp
## spcsPnspndr -0.715
## dfromtop    -0.309  0.221
## spcsPpndrs:  0.182 -0.309 -0.588
```

```
# Using Anova from car, we get p-values for the main effects.
library(car)
Anova(lme1)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: LMA
##                    Chisq Df Pr(>Chisq)
## species          147.475  1     <2e-16 ***
## dfromtop          86.832  1     <2e-16 ***
## species:dfromtop   2.531  1     0.1116
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We now conclude that `LMA` decreases significantly with `dfromtop`. Compare this with the fixed-effects model we started with:

```
summary(lm1)
```

```
##
## Call:
## lm(formula = LMA ~ species + dfromtop + species:dfromtop, data = pref)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -80.047 -21.737  -2.908  17.231 109.207
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      164.0749     4.5800  35.824   <2e-16 ***
## speciesPinus ponderosa           125.2444     6.2581  20.013   <2e-16 ***
## dfromtop                          -0.3062     0.4406  -0.695   0.4878
## speciesPinus ponderosa:dfromtop   1.1855     0.6923   1.712   0.0881 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 31.34 on 245 degrees of freedom
## Multiple R-squared:  0.8211,Adjusted R-squared:  0.8189
## F-statistic: 374.9 on 3 and 245 DF,  p-value: < 2.2e-16
```

Ignoring the tree-to-tree variance thus resulted in drawing the wrong conclusion from our data. When we accounted for this variation with a mixed-effects model, we did find a significant overall relationship between LMA and dfromtop. The reason for this discrepancy is that the large variation in intercepts between the individual trees masked the relationship between the two variables within individuals.

Finally, we may be interested in quantifying the variation between individuals in terms of the intercept and slope. These are the standard deviations of the random effects, and can be extracted with the `VarCorr` function (it is also shown in the `summary` statement of the mixed-effects model).

```
# Standard deviation of the intercept and slope ('dfromtop') between individuals,
# as estimated from the random effects.
VarCorr(lme2)
```

```
##  Groups   Name        Std.Dev. Corr
##  ID       (Intercept) 31.36023
##           dfromtop     0.47658 -0.304
##  Residual             17.96337
```

Try this yourself Try using `ranef` on `lme1` and `lme2`; the first will show intercepts for each of the random effects (trees), while the second will show both estimated intercepts and slopes for the random effects. Repeat the above example looking at the relationship between `narea` (leaf nitrogen per unit area), `dfromtop`, and `species`. Fitting a random intercept and slope may result in a convergence error, suggesting a poorly fitting model. If this happens, just ignore this step.

1.3 Example: blocked designs

Blocked designs are often used in field experiments to account for known or suspected environmental gradients at the study site. By blocking the experimental design, the effect of the environmental gradient can be separated from the effect of the treatment(s) of interest increasing the ability to detect significant effects. Let's look at the effects of herbicide and profile on soybean litter decomposition as a function of agricultural management (herbicide usage), microenvironment, and time. In this experiment, herbicide treatments were applied at the level of whole plots, with both treatments represented within each of four blocks. Both levels of variety and profile were each represented within each plot, with six replicates of each treatment added to each plot. The data description can be found in Section A.19.

The following code prepares the dataset for analysis, and produces Fig. 1.3.

```
# Read data
litter <- read.csv("masslost.csv")

# Make sure the intended random effects (plot and block) are factors
litter$plot <- as.factor(litter$plot)
litter$block <- as.factor(litter$block)

# Represent date as number of days since the start of the experiment
library(lubridate)
litter$date <- as.Date(mdy(litter$date))
litter$date2 <- litter$date - as.Date("2006-05-23")

# Quickly visualize the data to look for treatment effects
```

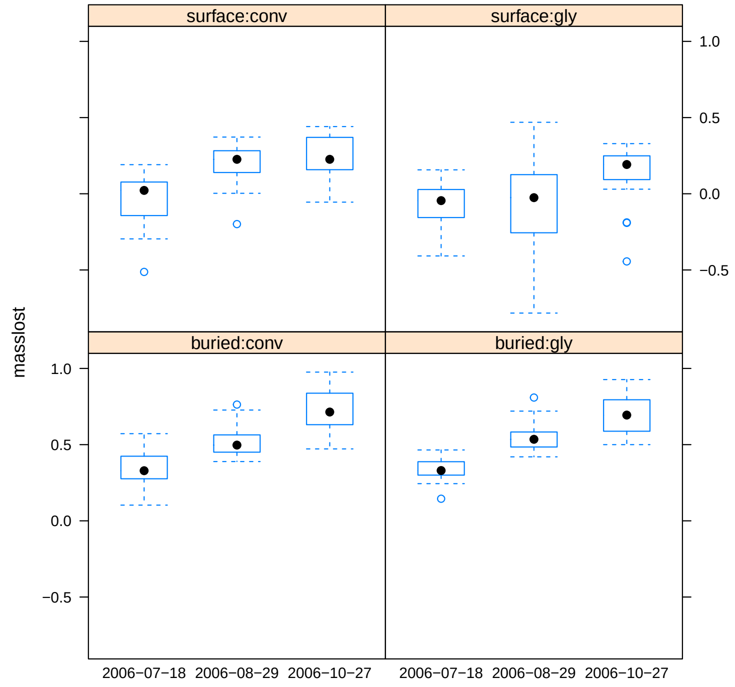

Figure 1.3: Proportion of litter mass lost from bags during field incubation as a function of microenvironment and herbicide program.

```
library(lattice)
bwplot(masslost ~ factor(date) | profile:herbicide, data=litter)
```

From inspecting Fig. 1.3, the buried litter appears to be decomposing faster than the surface litter (masslost is higher for buried compared to surface). If there are effects of herbicide (gly vs. conv), they are not immediately clear from the figure.

The blocking can be treated as a fixed effect or random effect. However, the design is unbalanced because some litter bags were lost, resulting in a variable number of litter bags recovered from each treatment. This affects the calculation of sums of squares, which vary depending on the order the terms are introduced to the model. It is therefore more appropriate to treat the blocking factor as a random effect, and use a mixed-effects model.

We first fit a simple linear model which ignores some details of the experimental design, and use block as a fixed effect. It is often very useful to start with a linear model, perhaps on subsets of the data, to gradually try to make sense of the data.

```
# Count the data to confirm that the design is unbalanced (ignore blocks for brevity)
ftable(xtabs(~ date2 + profile + herbicide, data=litter))

##                  herbicide conv gly
## date2 profile
## 56    buried                 22  22
##       surface                21  21
## 98    buried                 23  23
##       surface                20  20
## 157   buried                 19  16
##       surface                18  21
```

```
# Simple linear model with 'herbicide' as the first predictor in the model,
m1fix <- lm(masslost ~ date2 + herbicide * profile + block, data = litter)
anova(m1fix)
```

```
## Analysis of Variance Table
##
## Response: masslost
##                    Df  Sum Sq Mean Sq  F value    Pr(>F)
## date2               1  3.1696  3.1696 122.0249 < 2.2e-16 ***
## herbicide           1  0.4327  0.4327  16.6579 6.113e-05 ***
## profile             1 13.7225 13.7225 528.3018 < 2.2e-16 ***
## block               3  0.4674  0.1558   5.9987 0.0005891 ***
## herbicide:profile   1  0.3207  0.3207  12.3475 0.0005284 ***
## Residuals         238  6.1820  0.0260
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# ... or listing 'profile' first in the model.
m2fix <- lm(masslost ~ date2 + profile * herbicide + block, data = litter)
anova(m2fix)
```

```
## Analysis of Variance Table
##
## Response: masslost
##                    Df  Sum Sq Mean Sq  F value    Pr(>F)
## date2               1  3.1696  3.1696 122.0249 < 2.2e-16 ***
## profile             1 13.8335 13.8335 532.5749 < 2.2e-16 ***
## herbicide           1  0.3217  0.3217  12.3848 0.0005184 ***
## block               3  0.4674  0.1558   5.9987 0.0005891 ***
## profile:herbicide   1  0.3207  0.3207  12.3475 0.0005284 ***
## Residuals         238  6.1820  0.0260
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The sums of squares and p-values differ for `profile` and `herbicide` across the two fits (although they are still highly significant). Note that the order of the variables entered in the model matters because each next term is tested against a model that includes *all terms preceding it* (so-called Type-I tests). These standard tests with `anova` are sequential tests, which is perhaps not the most intuitive behaviour.

In many cases it is more intuitive to use so-called Type-II tests, in which each main effect is tested against a model that includes *all other terms*. We can use `Anova` (from the `car` package) to do this.

```
library(car)
Anova(m1fix)
```

```
## Anova Table (Type II tests)
##
## Response: masslost
##                    Sum Sq  Df  F value    Pr(>F)
## date2              3.7056   1 142.6601 < 2.2e-16 ***
## herbicide          0.3330   1  12.8198 0.0004157 ***
## profile           13.6594   1 525.8732 < 2.2e-16 ***
## block              0.4933   3   6.3311 0.0003793 ***
## herbicide:profile  0.3207   1  12.3475 0.0005284 ***
## Residuals          6.1820 238
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(m2fix)

## Anova Table (Type II tests)
##
## Response: masslost
##                     Sum Sq  Df  F value    Pr(>F)
## date2               3.7056   1 142.6601 < 2.2e-16 ***
## profile            13.6594   1 525.8732 < 2.2e-16 ***
## herbicide           0.3330   1  12.8198 0.0004157 ***
## block               0.4933   3   6.3311 0.0003793 ***
## profile:herbicide   0.3207   1  12.3475 0.0005284 ***
## Residuals           6.1820 238
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If the data were in fact balanced, the sequential and Type-II tests would be identical.

We also have not yet accounted for the fact that multiple litter bags were placed within single plots and that the 'herbicide' treatment was applied at the level of the plots, not the individual bags, which further complicates the analysis. Treating `block` and `plot` as random effects addresses both the imbalance and the hierarchical nature of the design.

In this example, we specify the nested nature of the data (plots within blocks) in the formula for the random effects as `(1|block/plot)`.

```
# fit model with random effects, plots nested within blocks
m1 <- lmer(masslost ~ date2 + herbicide * profile + (1|block/plot),
           data = litter)
Anova(m1)

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: masslost
##                      Chisq Df Pr(>Chisq)
## date2              142.525  1  < 2.2e-16 ***
## herbicide           12.749  1  0.0003561 ***
## profile            526.246  1  < 2.2e-16 ***
## herbicide:profile   12.187  1  0.0004813 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# As you can see for yourself with anova(m1, m2), plot explains
# essentially zero variance.
m2 <- lmer(masslost ~ date2 + herbicide * profile + (1|block),
           data = litter)
Anova(m2)

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: masslost
##                      Chisq Df Pr(>Chisq)
## date2              142.525  1  < 2.2e-16 ***
## herbicide           12.749  1  0.0003561 ***
## profile            526.246  1  < 2.2e-16 ***
## herbicide:profile   12.187  1  0.0004813 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

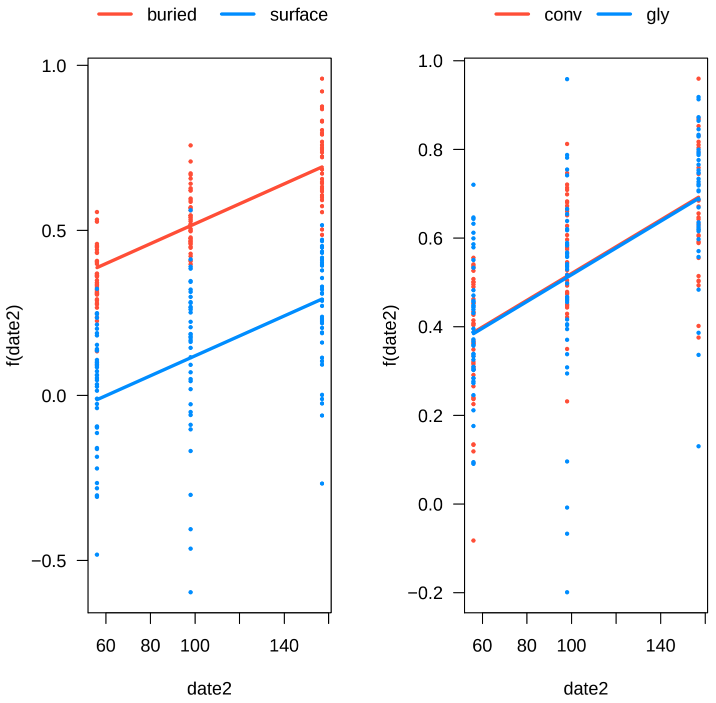

Figure 1.4: Visualization of the fixed effects of the mixed-effects model fit to the litter decomposition data.

Note that in the model, we used `date2` as a numeric variable, which assumes that the relationship between `masslost` and `date2` is more or less linear. Figure 1.4 shows that this is a resonable assumption However, in the case where you have a timeseries where no transformation exists to linearize the relationship, you will have to represent your time variable as a factor. We return to this issue in Section 1.5, after we treat another repeated measures example where time was continuous in Section 1.5.

Finally we visualize the fit, to make sense of the fitted coefficients, and to make sure we draw the right conclusions as to the direction of the significant effects. As we saw in Chapter 7, we can use `visreg` to quickly visualize the fitted model.

```
library(visreg)

# Because we have three fixed effects, we can make two plots to visualize all combinations.
par(mfrow=c(1,2))
visreg(m2, "date2", by="profile", overlay=TRUE)
visreg(m2, "date2", by="herbicide", overlay=TRUE)
```

1.3.1 More about p-values

As we mentioned at the top of this chapter, p-values in `lme4` are a bit controversial. We here used the `Anova` function, but another option is to use the `LMERConvenienceFunctions` package. This package includes functions to estimate of p-values from *conservative* and *liberal* assumptions about random effect degrees of freedom, and thus gives a range of possible p-values.

```
library(LMERConvenienceFunctions)

# To use the function below, we must fit with ML, not REML.
```

```
m2ml <- update(m2, REML=FALSE)

# calculate upper- and lower-bounds on p-values
pamer.fnc(m2ml)
```

```
##                   Df  Sum Sq Mean Sq  F value upper.den.df upper.p.val
## date2              1  3.1860  3.1860 124.7135          241      0e+00
## herbicide          1  0.4439  0.4439  17.3742          241      0e+00
## profile            1 13.6726 13.6726 535.2013          241      0e+00
## herbicide:profile  1  0.3153  0.3153  12.3406          241      5e-04
##                   lower.den.df lower.p.val expl.dev.(%)
## date2                      237       0e+00      13.1139
## herbicide                  237       0e+00       1.8269
## profile                    237       0e+00      56.2776
## herbicide:profile          237       5e-04       1.2976
```

Each main effect and interaction has two p-values: one assuming that each random effect accounts for one degree of freedom (`lower.p.val`) or no degrees of freedom (`upper.p.value`). The 'true' p-value will be somewhere in between these two bounds.

In this particular case, the results are highly significant (both lower and upper p-values are very small) because the effect size is quite large, but this will not always be the case.

Try this yourself Use `pamer.fnc` on the model above that contains random effects for both `Block` and `Plot`. What effect does this have on the degrees of freedom and p-value calculations? Why?

Another approach is to evaluate the importance of a term by comparing models that contain or do not contain that term using likelihood ratio tests with `anova` (p-values are approximate since likelihood ratios don't quite fit a chi-square distribution) or model selection based on `AIC` (the lower the better).

```
# remove the interaction term from the model
m2.int <- lmer(masslost ~ date2 + herbicide + profile + (1|block), data = litter)

# compare the two models
# Note that anova() will refit the models with ML (not REML) automatically,
# this is necessary when comparing models with different fixed or random effects terms.
anova(m2ml, m2.int)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: litter
## Models:
## m2.int: masslost ~ date2 + herbicide + profile + (1 | block)
## m2ml: masslost ~ date2 + herbicide * profile + (1 | block)
##        Df     AIC     BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## m2.int  6 -173.68 -152.64 92.838  -185.68
## m2ml    7 -183.70 -159.16 98.848  -197.70 12.02      1  0.0005263 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(m2ml, m2.int)
```

```
##        df       AIC
## m2ml    7 -183.6958
## m2.int  6 -141.5371
```

The model that includes the interaction provides the much better model fit. We can tell this by the significant p-value from the anova result and by the lower AIC score for the model that includes an interaction. When an interaction is significant, the automatic follow-up question is 'what is the source of this interaction?'. Again inspecting Fig. 1.4, it appears bthat the herbicide treatments affected decomposition differently, but only on the surface of the soil. To further understand the nature of the interactions, it is useful to combine variables into a single variable, as the following example illustrates.

```r
# Create a new variable containing the combinations of herbicide and profile.
# This new variable will have 4 levels
litter$combtrt <- paste(litter$herbicide, litter$profile, sep='-')

# Lump all observations for which the bags were buried into a single level,
# we now have just three levels in the new combined variable.
litter$combtrt[litter$profile == 'buried'] <- 'buried'

# Make this new variable into a factor
litter$combtrt <- as.factor(litter$combtrt)

# Fit a models using this new factor (3 levels) and a model without herbicide (2 levels)
m3 <- lmer(masslost ~ date2 + combtrt + (1|block), data = litter, REML=FALSE)
m3.herb <- lmer(masslost ~ date2 + profile + (1|block), data = litter, REML=FALSE)

# Compare the models by AIC (lower is 'better')
AIC(m2ml, m3, m3.herb)
```

```
##         df       AIC
## m2ml     7 -183.6958
## m3       6 -185.6860
## m3.herb  5 -163.6908
```

The model with the lowest AIC is m3, which is the model describing the relationship where herbicide affected decomposition rates only at the soil surface (because for that model, we combined all 'buried' litter samples into one level, regardless of the herbicide application).

1.4 Example: repeated measures

This example shows a very common use of mixed-effects models in repeated measurements. The basic idea is that when you have measurements on the same individuals (or plots, or some other unit) over time, you cannot treat the measurements as independent because that would be pseudo-replication, inflation of your sample size, and anti-conservative conclusions about significant effects.

We use data from the Hawkesbury Forest Experiment irrigation by fertilisation experiment (HFEIF, see Section A.16 for description of the data). In this experiment, sixteen plots of 72 *Eucalyptus saligna* trees were remeasured 20 times for height and diameter (although on a number of dates, not all trees were measured). Four treatments were applied (control, irrigated, fertilised, irrigated + fertilised). We ask in the following example whether tree height differs by treatment.

It is important that you recognize that the experimental unit in this example is the plot, not the tree, because the treatments were applied at a plot level. We therefore have to take into account the fact that trees are nested in plots, to avoid pseudoreplication.

```r
# Read data, make proper date and make sure the intended factor variables are factors.
hfeif <- read.csv("HFEIFbytree.csv")
hfeif$Date <- as.Date(hfeif$Date)

# Make sure plot number (plotnr) is a factor; it is read in as a numeric variable.
```

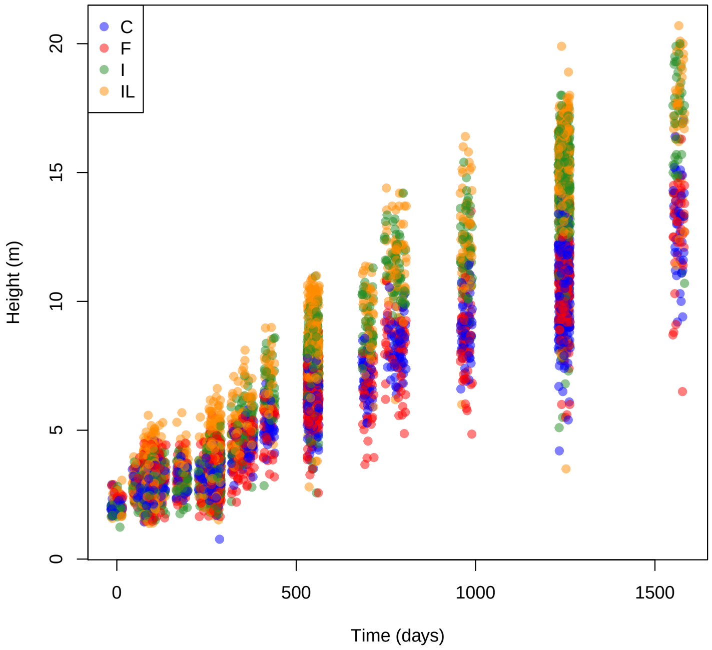

Figure 1.5: Plot tree height over time for the HFE irrigation x fertilisation experiment.

```
hfeif$plotnr <- as.factor(hfeif$plotnr)

# Days since start of experiment
# The as.numeric statement converts this into a simple numeric variable
hfeif$Time <- as.numeric(with(hfeif, Date - min(Date)))
```

Before we do anything, always explore the data with a few simple figures. Here we show tree height over time, colored by treatment (Fig. 1.5). The data show some separation between at least some of the treatments over time. Also note that the increase in height over time is perhaps not exactly linear, but we will ignore this in the remainder of the example (and further note that no straightforward transformation exists in this case).

Note the use of `jitter` in the example below, this adds some random noise to the Time variable to avoid excessive overlap of data points on each Date. We also use `sample` to randomly reorder the rows of the dataset to avoid the final treatment in the dataset to be plotted on top (this way, it is easier to see treatment differences).

The following code produces Fig. 1.5.

```
# For the alpha() function (transparency)
library(scales)

# Set colours, transparent
palette(alpha(c("blue","red","forestgreen","darkorange"), 0.5))
with(hfeif[sample(nrow(hfeif)),],
     plot(jitter(Time,3), height, col=treat, pch=19,
                 xlab="Time (days)", ylab="Height (m)"))
legend("topleft", levels(hfeif$treat), pch=19, col=palette())
```

Again, the reason we want to use mixed-effects models in this case is because we want to use the correct number of degrees of freedom to test for the treatment effect. If you are not sure what that should be, let's start with a linear model on the data from just one date, when we have averaged the data by plot (our experimental unit). In this case we can simply use `lm`, as follows.

The following example produces Fig. 1.6.

```
# Take subset of data at last Date
hfeif_last <- subset(hfeif, Date == max(Date))

# Average all variables by plot (and include the 'treat' factor variable in the result)
library(doBy)
hfeif_last_plot <- summaryBy(. ~ Date + plotnr, data=hfeif_last,
                             FUN=mean, na.rm=TRUE,
                             id=~treat, keep.names=TRUE)

# Linear model with treatment only.
lm_last <- lm(height ~ treat, data=hfeif_last_plot)

# Note that height is highly significant, and that we use 3 numerator df
# to test for treatment effects
anova(lm_last)

## Analysis of Variance Table
##
## Response: height
##           Df Sum Sq Mean Sq F value    Pr(>F)
## treat      3 74.160 24.7201  19.137 7.226e-05 ***
## Residuals 12 15.501  1.2917
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# A quick visualization of the fitted model shows much taller trees
# in I and IL,
library(visreg)
visreg(lm_last, "treat", xlab="Treatment", ylab="Tree height (m)")
```

The above shows 3 numerator degrees of freedom in the F-test, which makes sense because we have 4 levels of our treatment (thus df = 4 - 1). You can do no such simple check for the denominator degrees of freedom, but it's a useful check nonetheless.

To account for the repeated measures nature of the data as well as the fact that the experimental unit is the plot, not the tree, all we need is to specify the plot as the random effect. We will fit two models, one without and one with the interaction between `Time` and `treat`, and again use `Anova` (from the `car` package) to test for significant effects.

Note that we specify `Time` as a numeric variable, which assumes that the relationship between `height` and `Time` is more or less linear, which according to Figure 1.5 is a reasonable assumption. We return to this issue in Section 1.5.

```
# Effect of treatment on intercept only.
lmeif1 <- lmer(height ~ treat + Time + (1|plotnr), data=hfeif)
Anova(lmeif1)

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: height
##           Chisq Df Pr(>Chisq)
```

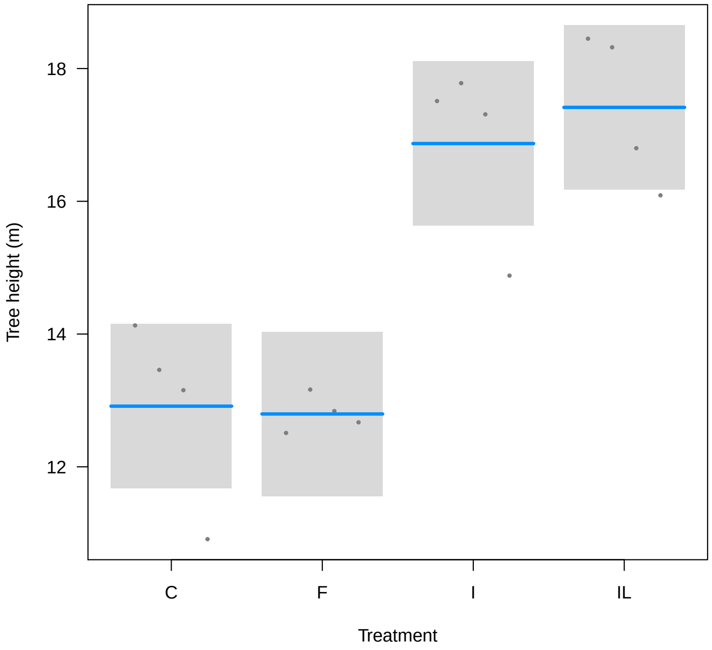

Figure 1.6: Simple visualization of fitted linear model (with lm) of tree height on the last date of the HFE IF data.

```
## treat    27.752  3  4.095e-06 ***
## Time  48339.942  1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Effect of treatment on intercept and slope (i.e. main effect + interaction)
lmeif2 <- lmer(height ~ treat*Time + (1|plotnr), data=hfeif)
Anova(lmeif2)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: height
##                 Chisq Df Pr(>Chisq)
## treat          27.389  3  4.879e-06 ***
## Time        72342.647  1  < 2.2e-16 ***
## treat:Time   3208.442  3  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# A likelihood ratio tests shows the interaction is highly significant
anova(lmeif1, lmeif2)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: hfeif
## Models:
## lmeif1: height ~ treat + Time + (1 | plotnr)
## lmeif2: height ~ treat * Time + (1 | plotnr)
```

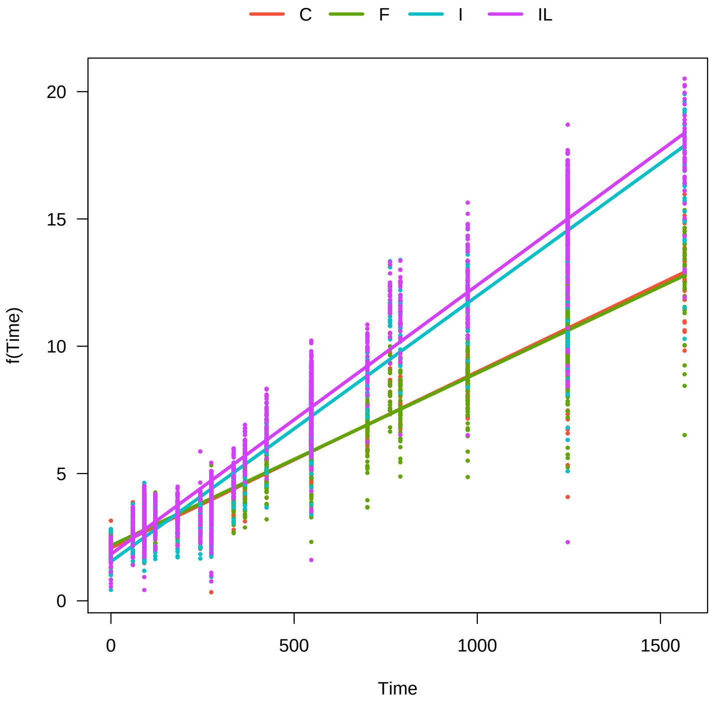

Figure 1.7: Visualized effect of Time on height, by treatment for the HFE IxF dataset, fitted with a linear mixed-effects model.

```
##        Df   AIC   BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmeif1  7 22654 22702 -11320    22640
## lmeif2 10 20054 20122 -10017    20034 2605.9      3  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that in the above, we have correctly used 3 numerator degrees of freedom to test for the effect of treatment on height. The individual `Anova` statements summarize the significance of the fixed effects in each model, whereas the `anova` of the two models uses a likelihood-ratio test on the two models. In this case, it effectively tests for significance of the interaction (because the only difference between the two models was the inclusion of the `treat` by `Time` interaction in `lmeif2`). The interaction is overwhelmingly significant.

The final step is to try to understand this interaction, how large is the effect size, and which direction does it point? The time by treatment interaction is significant, but in which way? It is never sufficient in an analysis to state that an interaction was 'significant', we must make more sense of it. One simple approach is to use the `visreg` package to visualize the fit (see Fig. 1.7).

```
library(visreg)
visreg(lmeif2,"Time", by="treat", overlay=TRUE)
```

In this case it is abundantly clear that irrigated (I) and irrigated + fertilised (IL) have a steeper slope of height with Time (that is, they have a faster height growth), compared to control (C) and fertilised (F). If there was no significant interaction (or a small effect size), the lines would be parallel to each other.

We can further look at the p-values for the individual effects (slopes and intercepts by treatment). Note that p-values in the `summary` statement are only computed if we have loaded the `lmerTest` package before fitting the model. Consider this example,

```
# Loading this package first affects both summary and anova methods
library(lmerTest)

##
## Attaching package:  'lmerTest'

## The following object is masked from 'package:lme4':
##
##     lmer

## The following object is masked from 'package:stats':
##
##     step

# ... we must refit the model after loading lmerTest
lmeif2 <- lmer(height ~ treat*Time + (1|plotnr), data=hfeif)

# Print just the coefficients table from the summary
summary(lmeif2)$coefficients

##                  Estimate   Std. Error         df     t value     Pr(>|t|)
## (Intercept)   2.073764669 2.992535e-01   12.32252   6.9297919 1.377196e-05
## treatF        0.093074081 4.233121e-01   12.33460   0.2198711 8.295684e-01
## treatI       -0.535359455 4.232604e-01   12.32858  -1.2648466 2.293128e-01
## treatIL      -0.246322580 4.232390e-01   12.32609  -0.5819940 5.710624e-01
## Time          0.006921244 6.399533e-05 6453.01255 108.1523211 0.000000e+00
## treatF:Time  -0.000141759 9.111223e-05 6453.02451  -1.5558723 1.197875e-01
## treatI:Time   0.003514427 9.095049e-05 6453.01005  38.6411001 0.000000e+00
## treatIL:Time  0.003645960 9.085136e-05 6453.02770  40.1310445 0.000000e+00
```

Try this yourself The `lmerTest` package also modifies the `anova` function, so that it calculates p-values for a fitted model with `lmer`. Compare `anova(lmeif2)` with `Anova(lmeif2)`, these will rarely be exactly the same as they use different methods to approximate the degrees of freedom of the random effects.

Looking at the interaction terms (treat:Time),the summary table shows that `treatF:Time` is not significantly different from the first level (`treatC:Time`), that is, there is no difference between fertilized and control in terms of the interaction with Time. But both irrigated (I) and irrigated + fertilised (IF) are highly significant, again, this comparison is in relation to the first level of the factor (control, C).

Although there is a significant main effect of treatment, none of the levels are actually different from the first (the control). This shows that the intercept itself is different from zero, but the treatments are not actually different in terms of the intercept. This makes sense, because seedlings were planted at time zero before any treatment was applied.

1.5 Repeated measures: is time numeric or factor?

In both examples in this chapter where we used time as a predictor in our models, we treated time as a continuous (numeric) variable. This was appropriate in both cases because the relationship between the dependent variable (`masslost` or `height`) showed a nearly linear relationship with time, allowing us to estimate and interpret an intercept and a slope of the variable with time. In the example with the tree height measurements, the slope of `height` with time can actually be interpreted as the height growth rate.

But there are many cases in which it would be more appropriate to use time as a factor variable. These

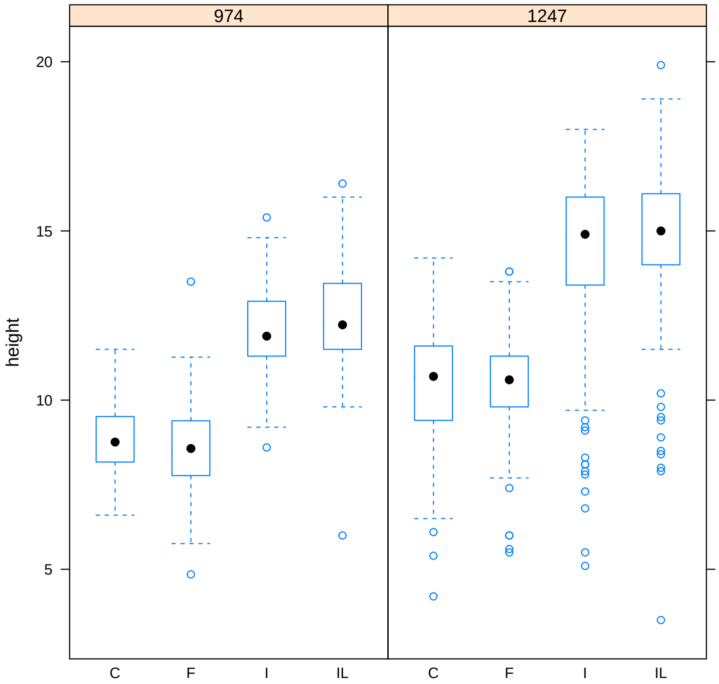

Figure 1.8: Boxplots of height vs. treatment and time (days since start of experiment, shown in the panel label) for a subset of the HFE IF data.

include cases where the relationship is highly non-linear and cannot be transformed, or you only have two or three dates of measurements. The example below shows a simple example for the latter case. (see also Fig. 1.8).

```
# A repeated measures example with only two dates of measurement.
# Though it is possible to have time as a continuous variable, it is much
# more useful to code it as a factor.

# We take a subset of hfeif.
hfeif2 <- subset(hfeif, Date %in% as.Date(c("2010-09-01","2011-06-01")))

# Convert Time to a factor
hfeif2$Time_fac <- as.factor(hfeif2$Time)

# As before, we can quickly use bwplot to inspect the data
library(lattice)
bwplot(height ~ treat | Time_fac, data=hfeif2)
```

Try this yourself Repeat the above example, but using the entire dataset (rather than a subset of the data for two dates). Inspect the model with `Anova`, and also look at the `summary` statement.

Let's fit the mixed-effects model on this small subset of the data to test whether a) treatment affects height, b) there is an effect of time on height, c) there is an interaction (i.e. height response to treatment depends on time).

```
# Fit the model
lmeif4 <- lmer(height ~ treat*Time_fac + (1|plotnr), data=hfeif2)

# Overall significance shows no interaction
Anova(lmeif4)
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: height
##                   Chisq Df Pr(>Chisq)
## treat           69.0334  3  6.874e-15 ***
## Time_fac       359.1303  1  < 2.2e-16 ***
## treat:Time_fac   8.4144  3    0.03818 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Try this yourself Use `visreg` to visualize the fit, and compare it to the box plots produced above.

1.6 Generalized linear mixed effects models

We have seen how to fit generalised linear models (section 7.6) and linear mixed models (above). Once you know how to fit these models in **R**, fitting GLMMs is fairly easy using the `glmer` function in the `lme4` package.

The EucFACE ground cover dataset (see Section A.20) contains estimates of plant and litter cover within the rings of the EucFACE experiment, evaluating forest ecosystem responses to elevated CO_2, on two dates. There are six rings (`Ring`), three for each treatment (`Trt`; ambient and elevated CO_2). Within each ring are four plots (`Plot`) and within each plot are four 1m by 1m subplots (`Sub`). Here we will test for an interaction between `Trt` and `Date` on ground cover measurements of `Forbes` (these are count data).

The following code produces Fig. 1.9, showing a plot of the raw groundcover data.

```
# read data and convert random effects to factors
eucface <- read.csv("eucfaceGC.csv")
eucface$Ring <- as.factor(paste(eucface$Ring, eucface$Trt, sep='-'))
eucface$Plot <- as.factor(eucface$Plot)
eucface$Sub <- as.factor(eucface$Sub)

# load packages
library(lme4)
library(lattice)

# A quick plot to visualize ground cover by Date and Ring.
# Colours represent 'Plot'.
xyplot(Forbes~Date|Ring,groups=Plot,data=eucface,pch=16,jitter.x=T)
```

Since the data are count data, it is usually appropriate to use the Poisson distribution. The following code fits a `glmer` with the `poisson` error family, and produces diagnostic plots are in Fig. 1.10.

To usual way to decide on the appropriate family is to inspect the diagnostic plots, especially a plot of residuals versus the fitted values. In the following, we fit the model three times, first with normal errors (using `lmer`), then with Poisson errors (the distribution we expect to be appropriate), and a Poisson distribution with square-root link function.

The following code produces three diagnostic plots (Fig. 1.10).

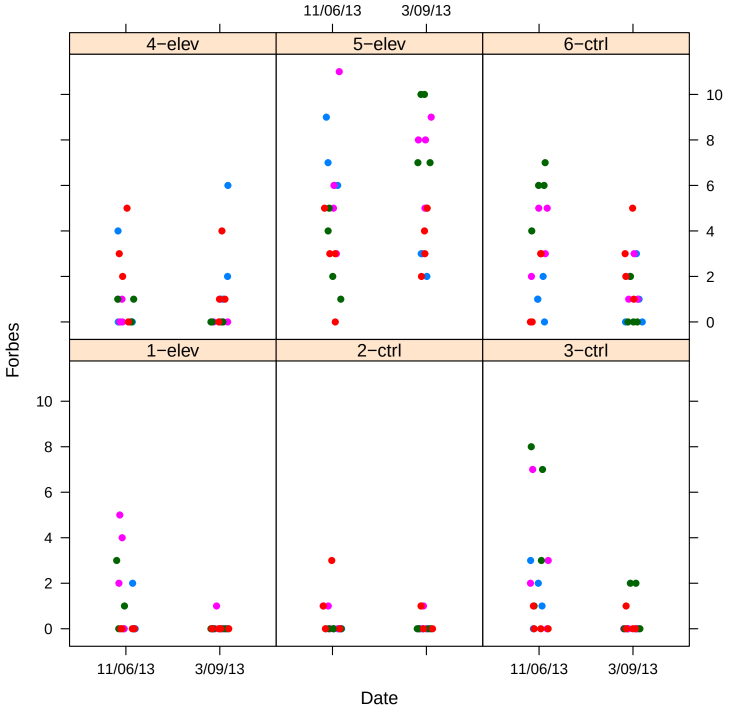

Figure 1.9: Counts of forbs within plant communities exposed to ambient or elevated carbon dioxide concentrations at two dates. Points represent estimates within subplots, subplots common to the same plot within each ring have a common colour.

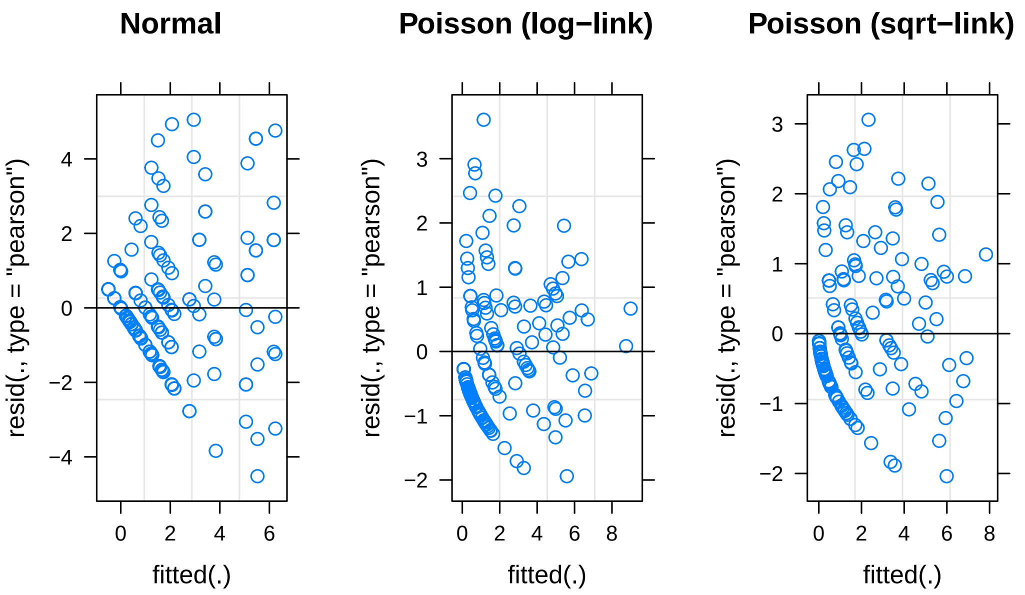

Figure 1.10: Diagnostic plots for three model fits of the EucFACE groundcover data.

```
forb.norm <- lmer(Forbes~Date*Trt+(1|Ring/Plot/Sub),
                  data=eucface)

forb.pois <- glmer(Forbes~Date*Trt+(1|Ring/Plot/Sub),
                   family=poisson, data=eucface)

forb.pois.sqrt <- glmer(Forbes~Date*Trt+(1|Ring/Plot/Sub),
                        family=poisson('sqrt'), data=eucface)

# The standard plot() on an (g)lmer object makes a 'grid-based'
# plot. To arrange plots side-by-side we have to use gridExtra, like so.
library(gridExtra)
p1 <- plot(forb.norm, main="Normal")
p2 <- plot(forb.pois, main="Poisson (log-link)")
p3 <- plot(forb.pois.sqrt, main="Poisson (sqrt-link)")
grid.arrange(p1,p2,p3,ncol=3)
```

Note that the syntax of `glmer` is identical to that of `lmer`, with the exception of the `family` argument.

Inspecting Fig. 1.10, the residuals look better for the models fit with the Poisson family compared to the gaussian error (as assumed by `lmer`). The residuals improve a bit more when using the `sqrt` link function in the Poisson family.

Note that the fit of the model may be improved further by using one of the other plant cover variables as a covariate or by incorporating other data from the site, but for the purpose of this chapter the fit is good enough.

Something else to consider when it comes to poisson errors is that overdispersion can result in the underestimation of error terms for the model coefficients. Overdispersion occurs when you have a large number of zeros in the data and/or an important predictor is not accounted for. Testing hypotheses from models where overdispersion is evident is dangerous as the probability of Type I error is increased. A good model fit should result in the ratio of residual deviance to degrees of freedom being close to one.

```r
# Calculate the residual deviance
sum(resid(forb.pois.sqrt, type='pearson')^2)
```

```
## [1] 171.408
```

```r
# model summary, note 'df.resid' in the model summary is 185
summary(forb.pois.sqrt)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( sqrt )
## Formula: Forbes ~ Date * Trt + (1 | Ring/Plot/Sub)
##    Data: eucface
##
##      AIC      BIC   logLik deviance df.resid
##    612.6    635.4   -299.3    598.6      185
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0391 -0.5597 -0.2764  0.4883  3.0589
##
## Random effects:
##  Groups          Name        Variance Std.Dev.
##  Sub:(Plot:Ring) (Intercept) 0.05631  0.2373
##  Plot:Ring       (Intercept) 0.12549  0.3542
##  Ring            (Intercept) 0.32556  0.5706
## Number of obs: 192, groups:  Sub:(Plot:Ring), 96; Plot:Ring, 24; Ring, 6
##
## Fixed effects:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)           1.1985     0.3556   3.371  0.00075 ***
## Date3/09/13          -0.5541     0.1132  -4.894 9.88e-07 ***
## Trtelev               0.1218     0.5022   0.243  0.80833
## Date3/09/13:Trtelev   0.3729     0.1604   2.325  0.02007 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Dt3/09/13 Trtelv
## Date3/09/13 -0.152
## Trtelev     -0.707  0.109
## Dt3/09/13:T  0.106 -0.710    -0.150
```

Overdispersion does not appear to be a problem here as the ratio is less than one. If it was, we could use a quasipoisson family as for GLMs (but unfortunately `glmer` does not support that family). It has also been suggested that including individual-level random effects in the model could alleviate the problem of overdispersion. We try this in the following example.

```r
# Create a variable for individual-level random effects
eucface$Ind <- as.factor(1:nrow(eucface))

# fit the model and look at the model summary
forb.pois.ind <- glmer(Forbes~Date*Trt+(1|Ring/Plot/Sub/Ind),
                       family=poisson(sqrt), data=eucface)
summary(forb.pois.ind)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( sqrt )
## Formula: Forbes ~ Date * Trt + (1 | Ring/Plot/Sub/Ind)
##    Data: eucface
##
##      AIC      BIC   logLik deviance df.resid
##      612      638     -298      596      184
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.7030 -0.5273 -0.2436  0.4350  2.5908
##
## Random effects:
##  Groups                Name        Variance Std.Dev.
##  Ind:(Sub:(Plot:Ring)) (Intercept) 0.06872  0.2621
##  Sub:(Plot:Ring)       (Intercept) 0.01199  0.1095
##  Plot:Ring             (Intercept) 0.12407  0.3522
##  Ring                  (Intercept) 0.32530  0.5703
## Number of obs: 192, groups:
## Ind:(Sub:(Plot:Ring)), 192; Sub:(Plot:Ring), 96; Plot:Ring, 24; Ring, 6
##
## Fixed effects:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)           1.1766     0.3564   3.302 0.000961 ***
## Date3/09/13          -0.5554     0.1266  -4.387 1.15e-05 ***
## Trtelev               0.1219     0.5031   0.242 0.808482
## Date3/09/13:Trtelev   0.3763     0.1788   2.104 0.035368 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Dt3/09/13 Trtelv
## Date3/09/13 -0.170
## Trtelev     -0.706  0.121
## Dt3/09/13:T  0.118 -0.710    -0.168
```

The random effects block indicates that a small amount of variance is accounted for by the individual level random effects. This is expected as we did not observe overdispersion.

We can use `Anova` function from the `car` package to calculate significance associated with the main effects and interaction. Note that is is only one way of many to calculate p-values, as we discussed above linear mixed models.

```
library(car)
Anova(forb.pois.ind)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: Forbes
##            Chisq Df Pr(>Chisq)
## Date     16.8793  1  3.983e-05 ***
## Trt       0.3659  1    0.54527
## Date:Trt  4.4273  1    0.03537 *
## ---
```

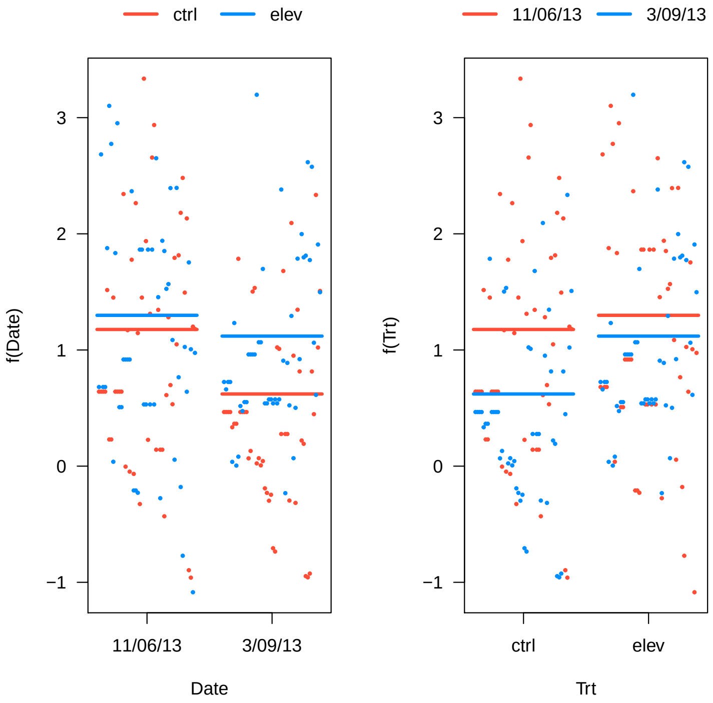

Figure 1.11: Model predictions, looking at treatment effects by date (left) and date effects by treatment (right).

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

There appears to be a significant interaction between `Date` and `Trt`. Again, when an interaction is significant we must dig deeper to understand the source of the interaction. Looking at Fig. 1.11, it appears that :

1. Forbs decreased in frequency between the two dates in the control but not in the elevated CO_2 treatment
2. Forb abundance was lower in the control than in the elevated CO_2 treatment on the second date, but not the first date.

```
# set up graphics window for two plots
par(mfrow=c(1, 2))

# plot model preditions and data
visreg(forb.pois.ind, 'Date', 'Trt', overlay=TRUE)
visreg(forb.pois.ind, 'Trt', 'Date', overlay=TRUE)
```

We can evaluate each of these hypotheses through model selection after combining treatment levels.

```
# create a three level factor that combines both dates in the 'elev' treatment
eucface$trtcomb.elev <- with(eucface, paste(Trt, Date, sep='-'))
eucface$trtcomb.elev[eucface$Trt == 'elev'] <- 'elev'
eucface$trtcomb.elev <- as.factor(eucface$trtcomb.elev)
levels(eucface$trtcomb.elev)

## [1] "ctrl-11/06/13" "ctrl-3/09/13"  "elev"

# create a three level factor that combines both treatments for the '11/06/13' sampling
eucface$trtcomb.date <- with(eucface, paste(Trt, Date, sep='-'))
```

```
eucface$trtcomb.date[eucface$Date == '11/06/13'] <- '11/06/13'
eucface$trtcomb.date <- as.factor(eucface$trtcomb.date)
levels(eucface$trtcomb.date)

## [1] "11/06/13"      "ctrl-3/09/13" "elev-3/09/13"

# fit model with three level combination factors
m2.elev <- glmer(Forbes ~ trtcomb.elev + (1|Ring/Plot/Sub/Ind),
                 family=poisson('sqrt'), data=eucface)

m2.date <- glmer(Forbes ~ trtcomb.date + (1|Ring/Plot/Sub/Ind),
                 family=poisson('sqrt'), data=eucface)

# fit models with only one main effect (Date or Trt)
m3.Trt <- glmer(Forbes~Date+(1|Ring/Plot/Sub/Ind), family=poisson(sqrt), data=eucface)
m3.Date <- glmer(Forbes~Trt+(1|Ring/Plot/Sub/Ind), family=poisson(sqrt), data=eucface)

# compare models (model with the lowest AIC is the most efficient at predicting the response)
AIC(forb.pois.ind, m2.elev, m2.date, m3.Trt, m3.Date)

##                df      AIC
## forb.pois.ind  8 611.9524
## m2.elev        7 611.9420
## m2.date        7 610.0109
## m3.Trt         6 612.6342
## m3.Date        6 627.4332
```

The three-level model in which date effects are estimated in the control treatment but not the elevated treatment (m2.elev) is not a substantial improvement over the fully factorial model (forb.pois.ind), so we do not consider it further. The three-level model in which treatment effects are estimated on the second date but not the first date (m2.date) has the lowest AIC score in comparison to the fully factorial model (forb.pois.ind) and to the two-level models that do not estimate an effect of treatment (m3.Trt) or date (m3.Date), so is the model that we select as the one that best predicts forb dynamics.

1.6.1 Logistic regression using mixed effects models

Something that wasn't mentioned regarding this EucFACE ground cover dataset is that vegetation was assessed at a maximum of 16 points within each subplot and, therefore, the maximum number of observations per plot is constrained. This does not affect our analysis of forb abundances because these are generally low (less than ten in almost all plots) and so interpreting these as count data is appropriate. This is not the case for other response variables. As an example, let's look at grass cover in Fig. 1.12.

```
# A quick plot to visualize ground cover by Date and Ring.
# Colours represent 'Plot'.
xyplot(Grass~Date|Ring,groups=Plot,data=eucface,pch=16,jitter.x=T)
```

The data are clearly bounded at both the lower and upper range. We can treat these as binomial distributed, with the presence or absence of grass assessed at each of the sixteen points within each subplot. Do do this, we use cbind to create a two-column response matrix indicating the number of presences and absences within each subplot, as we did for logistic regression in Section 7.5. We also specify the random effects as we did for the analysis of forb abundance above.

```
# fit model with binomial error distribution
grass.binom <- glmer(cbind(Grass, 16-Grass)~Date*Trt+(1|Ring/Plot/Sub),
                     family=binomial, data=eucface)
```

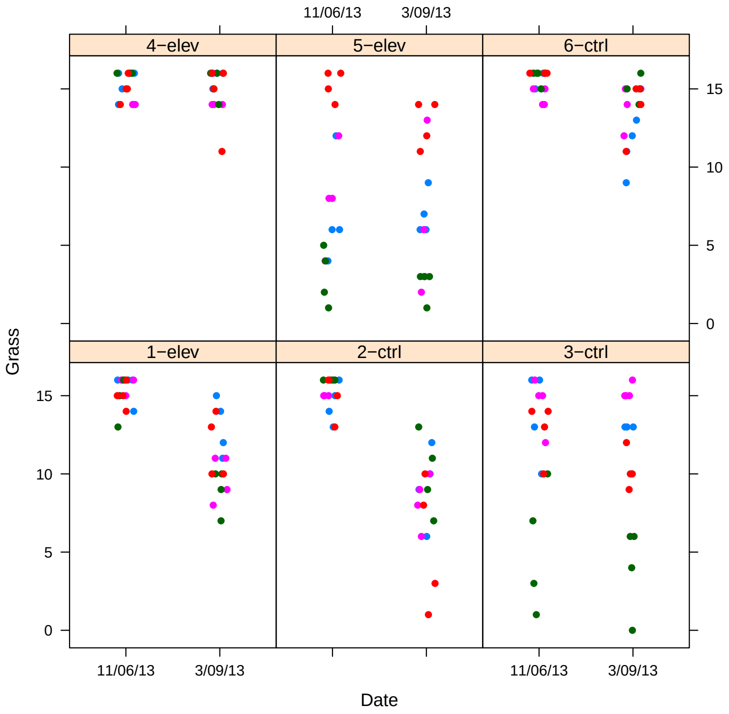

Figure 1.12: Grass cover within plant communities exposed to ambient or elevated carbon dioxide concentrations at two dates. Cover was assessed based on presence at each of sixteen locations within a subplot. Points represent estimates within subplots, subplots common to the same plot within each ring have a common colour.

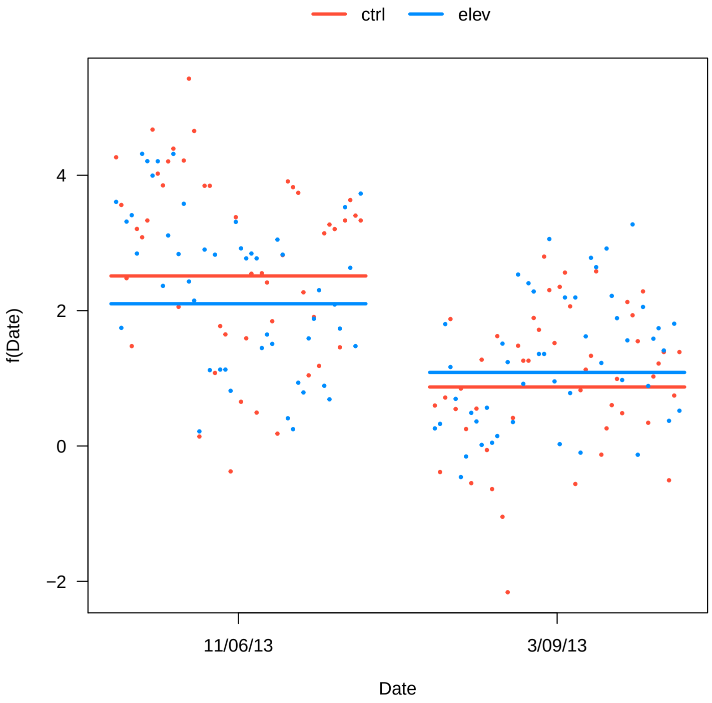

Figure 1.13: Model predictions, looking at treatment effects by date.

```
# test significance of main effects and interaction
Anova(grass.binom)

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: cbind(Grass, 16 - Grass)
##             Chisq Df Pr(>Chisq)
## Date     145.3645  1  < 2.2e-16 ***
## Trt        0.0000  1   0.997748
## Date:Trt   8.3183  1   0.003925 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The results suggest a highly significant effect of `date` and a signficant `date` by `treatment` interaction. The model predictions are shown in Fig. 1.13. A similar approach could be used to tease apart the source of the interaction as was used for forb abundances.

```
# plot model preditions and data
visreg(grass.binom, 'Date', 'Trt', overlay=TRUE)
```

1.7 Exercises

In the exercises, we use the following colour codes:

■ **Easy**: make sure you complete some of these before moving on. These exercises will follow examples in the text very closely.

♦ **Intermediate**: a bit harder. You will often have to combine functions to solve the exercise in two steps.

▲ **Hard**: difficult exercises! These exercises will require multiple steps, and significant departure from examples in the text.

1.7.1 PREF Canopy data

1. ♦ In the analysis of the `pref` data, use model selection (`AIC`, `anova`) to evaluate the importance of `species` and `dfromtop`.

1.7.2 Litter decomposition data

1. ♦ The `litter` data contain a factor (`variety`) describing whether the litter is derived from a genetically modified (`gm`) or conventional (`nongm`) soy variety. Plot the data to observe the effect of `variety`. Use `lmer` to test the effect of `variety`, in addition to the other significant variables, on litter decomposition.

1.7.3 EucFACE ground cover data

The file `eucfaceGC.csv` contains estimates of plant and litter cover within the rings of the EucFACE experiment, evaluating forest ecosystem responses to elevated CO2, on two dates; the data description can be found in Section A.20 (p. 233).

1. ♦ Convert the variables indicating the nested sampling design to factors, then use `glmer` in `lme4` to test for an interaction between `Trt` and `Date` on `Grass` and `Litter` cover. Grass cover represents a frequency across a maximum of 16 points within a quadrat (use the `binomial` family), while litter cover represents counts (use the `poisson` family).

2. ▲ Following on from exercise 3, generate subsets to determine the sources of the interactions (i.e., does the treatment effect differ between the two dates or does the date effect differ between the two treatments?).

Index