mixed_models_guide

Julia Piaskowski

8/3/22

Table of contents

Preface

This is a defaulf file accompanying a quarto book.

To learn more about Quarto books visit https://quarto.org/docs/books.

1 Introduction

This guide is focused on frequentist implementations of mixed models in R. If someone wants to write a Bayesian guide, please go for it! I'm not experience sufficiently in Bayesian to do this.

Each section contains the minimum to run a model, with more detail found at the later chapters. Unless I decide it makes more sense to include early materials.

A Tidymodels framework is used whenever possible because that is a promising avenue for making the syntax easier to write across packages.

2 Basic Models

3 Generalized Linear Mixed Models

```
library(dplyr)
```

```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(glmmTMB)
library(DHARMa)
```

```
This is DHARMa 0.4.6. For overview type '?DHARMa'. For recent changes, type news(package = 'I
```

3.1 Hurdle model

```
insect_exp <- read.csv("data/insect_count_data_glmm.csv")
```

plot: a unique number referring to each experimental unit

treatment: pesticidal treatment (6 different products)

row: plot position for row

col: plot positions for column or range

block: the blocking unit

insect__counts: response variable

sampling__date: dates when each experimental unit were evaluated for insect counts

```
head(insect_exp)
```

```
  plot treatment row column block insect_counts sampling_date
1  101         2   1      1     1             4       6/17/88
2  102         5   1      2     1             1       6/17/88
3  103         1   1      3     1             0       6/17/88
4  104         6   1      4     1             0       6/17/88
5  201         3   2      1     1             0       6/17/88
6  202         4   2      2     1             0       6/17/88
```

Two new variables created:

treatment: original variable treatment converted to a factor

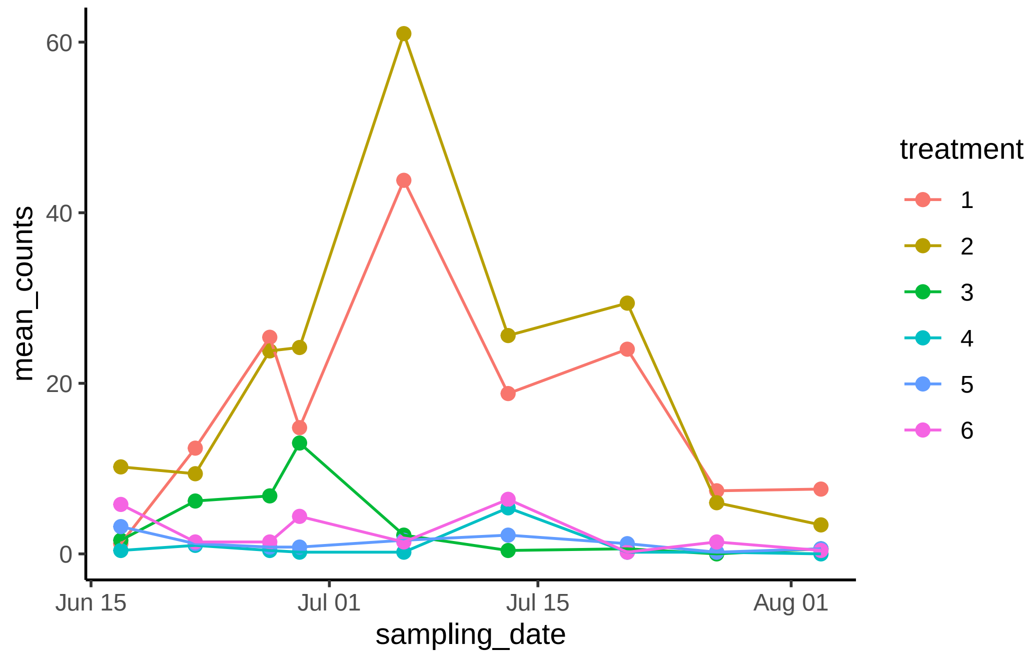
block: original variable block converted to a factor

Date: factor version of sampling_date

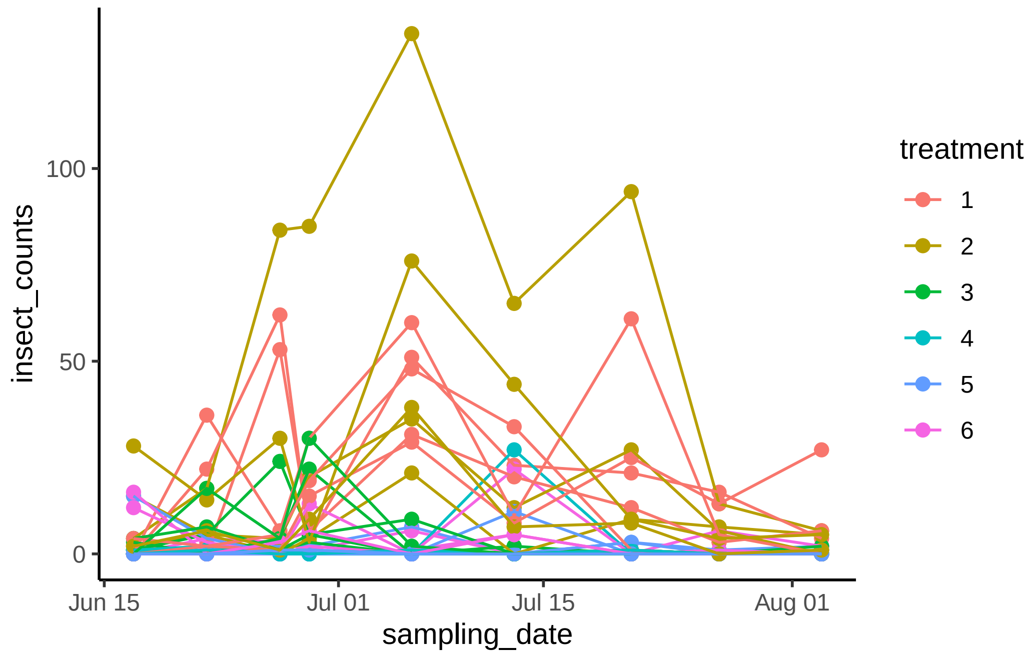
```
library(dplyr)

insect_exp <- insect_exp %>%
  mutate(block = as.factor(block),
         treatment = as.character(treatment),
         sampling_date = as.Date(sampling_date, format = "%m/%d/%y")) %>%
  mutate(Date = as.factor(sampling_date))
```

Visualise data

```
library(ggplot2)

insect_exp %>% group_by(sampling_date, treatment) %>%
  summarise(mean_counts = mean(insect_counts)) %>%
  ggplot(., aes(x = sampling_date, y = mean_counts, color = treatment)) +
    geom_point(size = 2) +
    geom_line() +
    theme_classic()
```



```
ggplot(insect_exp, aes(x = sampling_date, y = insect_counts, color = treatment, group = pl
  geom_point(size = 2) +
  geom_line() +
  theme_classic()
```



Model statement {[[[[FIX THIS - it's still written for alfalfa]]]]}

$$y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_k + a_l + b_m + c_n + \epsilon$$

where

μ = overall mean/intercept α_i = effect of the i^{th} pesticide treatment β_j = effect of the j^{th} block γ_k = effect of the k^{th} sampling date

To make things easier, the interactions between the fixed effects are not shown.

```
library(glmmTMB)

m1 = glmmTMB(
  insect_counts ~ treatment + Date + ar1(Date + 0|plot) + (1|block),
  ziformula = ~ treatment,
  data = insect_exp, na.action = na.exclude,
  family = nbinom2)
```

special correlation structure for correlated error terms `ar1()` (autoregressive 1).

There are several other specialized covariance structures implmented by glmmTMB. In general, repeated measures syntax follow this convention: `(time + 0 | grouping)`.

We can test other distributions

```
m2 <- update(m1, family = poisson)
```

```
Warning in fitTMB(TMBStruc): Model convergence problem; non-positive-definite
Hessian matrix. See vignette('troubleshooting')
```

```
Warning in fitTMB(TMBStruc): Model convergence problem; false convergence (8).
See vignette('troubleshooting')
```

```
m3 <- update(m1, family = nbinom1)
```

```
Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
NaN function evaluation
```

```
Warning in fitTMB(TMBStruc): Model convergence problem; non-positive-definite
Hessian matrix. See vignette('troubleshooting')
```

```
Warning in fitTMB(TMBStruc): Model convergence problem; false convergence (8).
See vignette('troubleshooting')
```

Fitting glmm is hard. Basic guidance on model fitting: https://glmmtmb.github.io/glmmTMB/articles/troublesl

```
diagnose(m2)
```

```
Unusually large Z-statistics (|x|>5):

    treatment3     treatment4     treatment5 Date1988-07-06 Date1988-07-13
     -6.616584     -15.016232     -10.313314       9.371860       6.381998
Date1988-08-03 zi~(Intercept)  zi~treatment4
     -5.415075     -22.156803      59.213969

Large Z-statistics (estimate/std err) suggest a *possible* failure of
the Wald approximation - often also associated with parameters that are
at or near the edge of their range (e.g. random-effects standard
deviations approaching 0).  (Alternately, they may simply represent
very well-estimated parameters; intercepts of non-centered models may
fall in this category.) While the Wald p-values and standard errors
listed in summary() may be unreliable, profile confidence intervals
(see ?confint.glmmTMB) and likelihood ratio test p-values derived by
```

comparing models (e.g. ?drop1) are probably still OK. (Note that the
LRT is conservative when the null value is on the boundary, e.g. a
variance or zero-inflation value of 0 (Self and Liang 1987; Stram and
Lee 1994; Goldman and Whelan 2000); in simple cases these p-values are
approximately twice as large as they should be.)

Non-positive definite (NPD) Hessian

The Hessian matrix represents the curvature of the log-likelihood
surface at the maximum likelihood estimate (MLE) of the parameters (its
inverse is the estimate of the parameter covariance matrix). A
non-positive-definite Hessian means that the likelihood surface is
approximately flat (or upward-curving) at the MLE, which means the
model is overfitted or poorly posed in some way. NPD Hessians are often
associated with extreme parameter estimates.

parameters with non-finite standard deviations:
(Intercept), Date1988-06-22, zi~treatment3, zi~treatment6,
theta_Date+0|plot.2, theta_1|block.1

recomputing Hessian via Richardson extrapolation. If this is too slow, consider setting chec

Hessian has complex eigenvalues

We would have used the smallest eigenvalues of the Hessian to determine
which components were bad but instead we got complex eigenvalues. (Not
really sure what to do with this ...)

```r
diagnose(m3)
```

Unusually large coefficients (|x|>10):

```
d~(Intercept)
    -26.35181
```

Large negative coefficients in zi (log-odds of zero-inflation),
dispersion, or random effects (log-standard deviations) suggest

```
unnecessary components (converging to zero on the constrained scale);
large negative and/or positive components in binomial or Poisson
conditional parameters suggest (quasi-)complete separation. Large
values of nbinom2 dispersion suggest that you should use a Poisson
model instead.


Unusually large Z-statistics (|x|>5):

treatment5
 -6.281128

Large Z-statistics (estimate/std err) suggest a *possible* failure of
the Wald approximation - often also associated with parameters that are
at or near the edge of their range (e.g. random-effects standard
deviations approaching 0).  (Alternately, they may simply represent
very well-estimated parameters; intercepts of non-centered models may
fall in this category.) While the Wald p-values and standard errors
listed in summary() may be unreliable, profile confidence intervals
(see ?confint.glmmTMB) and likelihood ratio test p-values derived by
comparing models (e.g. ?drop1) are probably still OK.  (Note that the
LRT is conservative when the null value is on the boundary, e.g. a
variance or zero-inflation value of 0 (Self and Liang 1987; Stram and
Lee 1994; Goldman and Whelan 2000); in simple cases these p-values are
approximately twice as large as they should be.)


Non-positive definite (NPD) Hessian

The Hessian matrix represents the curvature of the log-likelihood
surface at the maximum likelihood estimate (MLE) of the parameters (its
inverse is the estimate of the parameter covariance matrix).  A
non-positive-definite Hessian means that the likelihood surface is
approximately flat (or upward-curving) at the MLE, which means the
model is overfitted or poorly posed in some way. NPD Hessians are often
associated with extreme parameter estimates.


parameters with non-finite standard deviations:
d~(Intercept)
```

```
recomputing Hessian via Richardson extrapolation. If this is too slow, consider setting chec
```

```
Hessian has complex eigenvalues
```

```
We would have used the smallest eigenvalues of the Hessian to determine
which components were bad but instead we got complex eigenvalues. (Not
really sure what to do with this ...)
```

Summary info

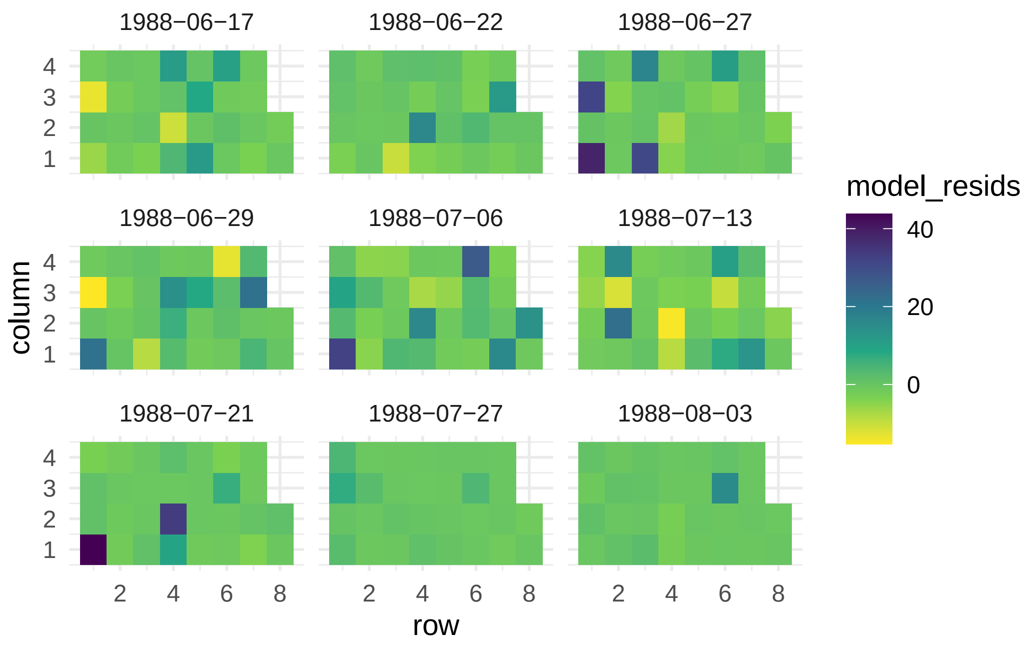
```
m1
```

```
Formula:
insect_counts ~ treatment + Date + ar1(Date + 0 | plot) + (1 |      block)
Zero inflation:                 ~treatment
Data: insect_exp
      AIC       BIC    logLik  df.resid
1298.7328 1385.0949 -625.3664       246
Random-effects (co)variances:

Conditional model:
 Groups Name           Std.Dev. Corr
 plot   Date1988-06-17 0.7748   0.49 (ar1)
 block  (Intercept)    0.3333

Number of obs: 270 / Conditional model: plot, 30; block, 5

Dispersion parameter for nbinom2 family (): 1.76

Fixed Effects:

Conditional model:
   (Intercept)      treatment2      treatment3      treatment4      treatment5
       2.39231        -0.04978        -1.53159        -2.75395        -2.50652
    treatment6  Date1988-06-22  Date1988-06-27  Date1988-06-29  Date1988-07-06
      -1.48975         0.24054         0.26618         0.62692         1.17067
Date1988-07-13  Date1988-07-21  Date1988-07-27  Date1988-08-03
       0.83442         0.19962        -0.96749        -1.11938

Zero-inflation model:
(Intercept)   treatment2   treatment3   treatment4   treatment5   treatment6
     -2.608       -1.200        1.568        2.607        1.542        2.134
```

Diagnostics

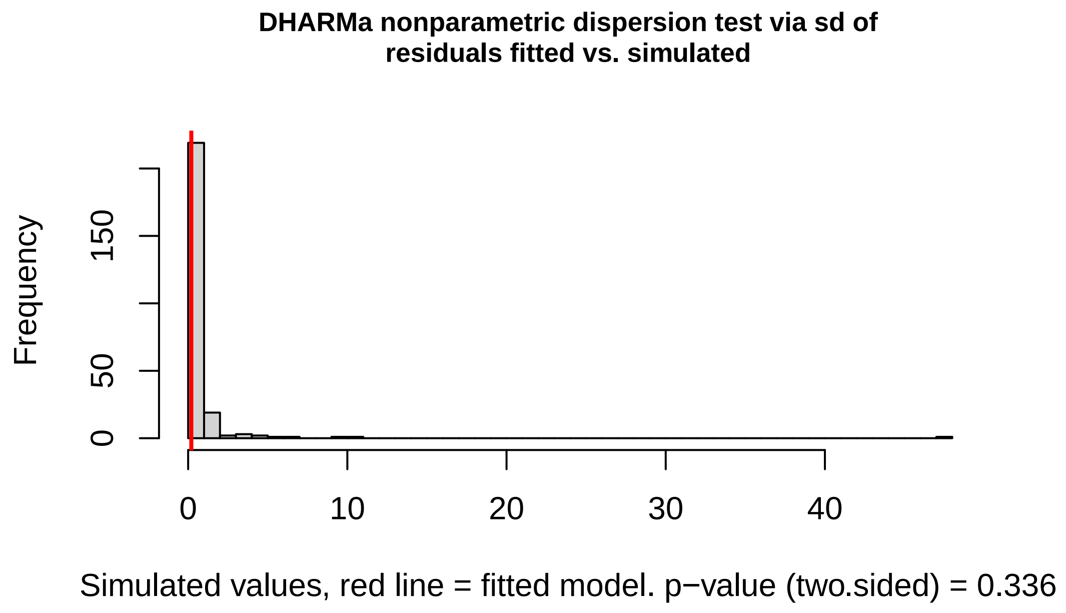
Look at residuals over space

```
insect_exp$model_resids <- residuals(m1)

ggplot(insect_exp, aes(x = row, y = column, fill = model_resids)) +
  geom_tile() +
  facet_wrap(facets = vars(Date), nrow = 3, ncol = 3) +
  scale_fill_viridis_c(direction = -1) +
  theme_minimal()
```



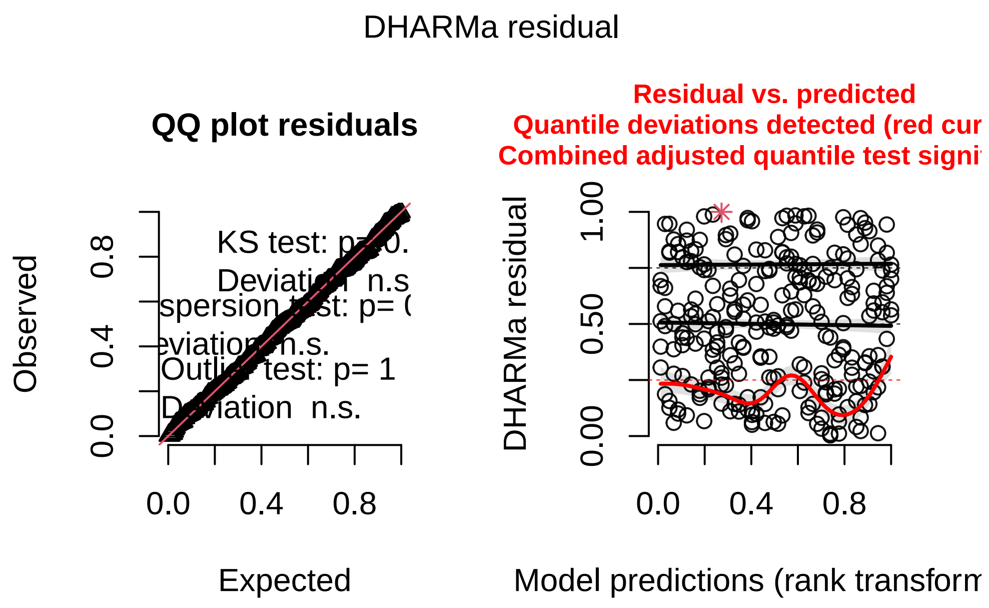
use **DHARMa** to conduct residual tests

```
simulated_resids <- simulateResiduals(m1)
testDispersion(simulated_resids)
```



```
	DHARMa nonparametric dispersion test via sd of residuals fitted vs.
	simulated

data:  simulationOutput
dispersion = 0.23324, p-value = 0.336
alternative hypothesis: two.sided
```

```
plot(simulated_resids)
```



ANOVA

```r
car::Anova(m1)
```

```
Analysis of Deviance Table (Type II Wald chisquare tests)

Response: insect_counts
           Chisq Df Pr(>Chisq)
treatment 54.358  5  1.769e-10 ***
Date      41.652  8  1.574e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

glmmTMB is compatible with **emmeans** and **effects**.

4 Special Conditions

4.1 Split plot with repeated measures

Main plot is "irrigation" and split plot is "mix".

```r
alfalfa_sp <- read.csv("data/alfalfa2021_data.csv")
library(dplyr)
```

```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

cut: a cutting (harvest) of alfalfa within a single growing season. This is a temporal unit for repeated measures analysis. There were three cuttings in total for that year and field. The dates are not known, but we cannot assume they are evenly spaced apart.

irrigation: irrigation treatment ("Full" or "Deficit")

plot: a unique number referring to each experimental unit

block: the blocking unit

yield: response variable

row: plot position for row

col: plot positions for column or range

```r
head(alfalfa_sp)
```

```
    cut irrigation plot block     mix    yield row col
1 First       Full 1101     1 50A+50O 221.0418   1   1
2 First       Full 1102     1 75A+25O 288.7987   1   2
3 First       Full 1103     1 50A+50F 466.7924   1   3
4 First       Full 1104     1 75A+25M 556.9506   1   4
5 First       Full 1105     1 50A+50M 422.9160   1   5
6 First       Full 1106     1 75A+25F 289.8350   2   1
```

Two new variables created:

rep: factor version of block (We should treat rep/block as a factor rather than an integer in modelling)

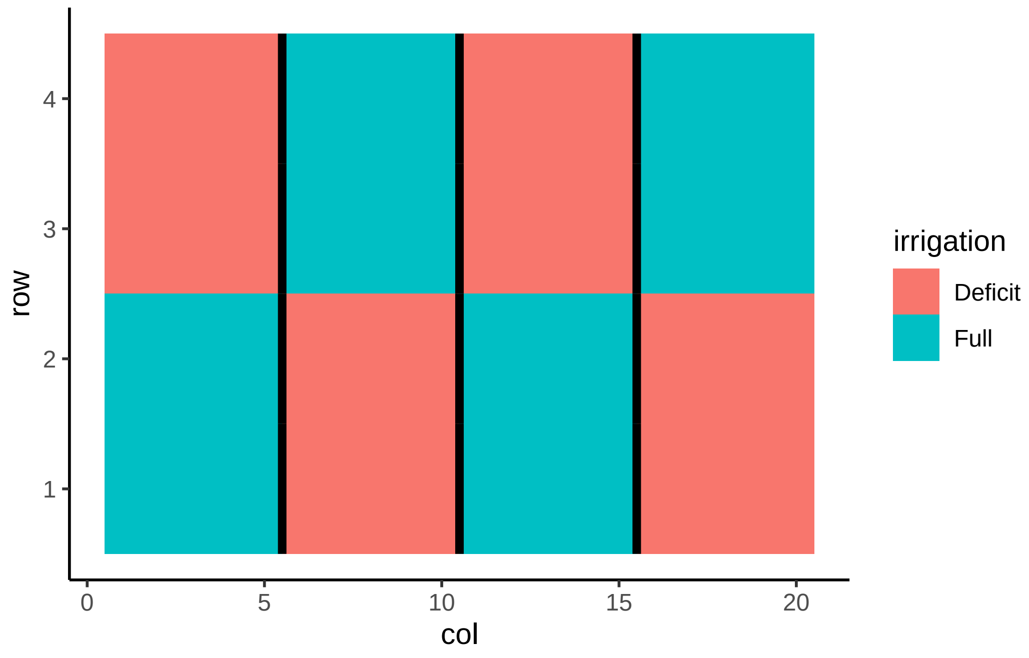
Cut: number version of cut where 1 is the first cutting. This is required by `nlme::lme` for specialized correlation structures.

```
alfalfa_sp <- alfalfa_sp %>%
  mutate(rep = as.factor(block)) %>%
  mutate(Cut = case_when(
    cut == "First" ~ 1L,
    cut == "Second" ~ 2L,
    cut == "Third" ~ 3L,
    is.na(cut) ~ NA_integer_))
```

Visualise data

```
library(ggplot2); library(desplot)

alfalfa_sp %>% filter(cut == "First") %>%

ggplot(aes(x = col, y = row)) +
  geom_raster(aes(fill = irrigation)) +
  geom_tileborder(aes(group = 1, grp = rep), lwd = 1.5) +
  theme_classic()
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```



Model statement

$$y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_k + a_l + b_m + c_n + \epsilon$$

where

μ = overall mean/intercept
α_i = effect of the i^{th} irrigation treatment
β_j = effect of the j^{th} planting mix treatment γ_k = effect of the k^{th} cutting [[need all those interactions]]

```
library(nlme)
```

```
Attaching package: 'nlme'


The following object is masked from 'package:dplyr':

    collapse
```

```
m1 <- lme(yield ~ mix*irrigation*cut,
          random = ~ 1|rep/irrigation/plot,
          data = alfalfa_sp)
```

use a special correlation structure for correlated error terms `corCompSymm()` is for compound symmetry. There are several other options in the **nlm** machinery (search "cor" for more options and details on the syntax). In general, repeated measures syntax follow this convention: `form = ~ time|grouping`. You can also use `1|group` and the observation order for each group will be. The default starting value (`value`) is zero, and if `fixed = FALSE` (the current nlme default), this value will be allowed to change during the model fitting process.

```
corstr <- corCompSymm(value = 0.3,
                      form = ~ cut|rep/irrigation/plot,
                      fixed = FALSE)
```

It's important that these two terms match after the "|" in the `random` and `form` arguments:

```
m1 <- lme(yield ~ mix*irrigation*cut,
          random = ~ 1|rep/irrigation/plot,
          data = alfalfa_sp)

corstr <- corCompSymm(value = 0.3,
                      form = ~ cut|rep/irrigation/plot,
                      fixed = FALSE)
```

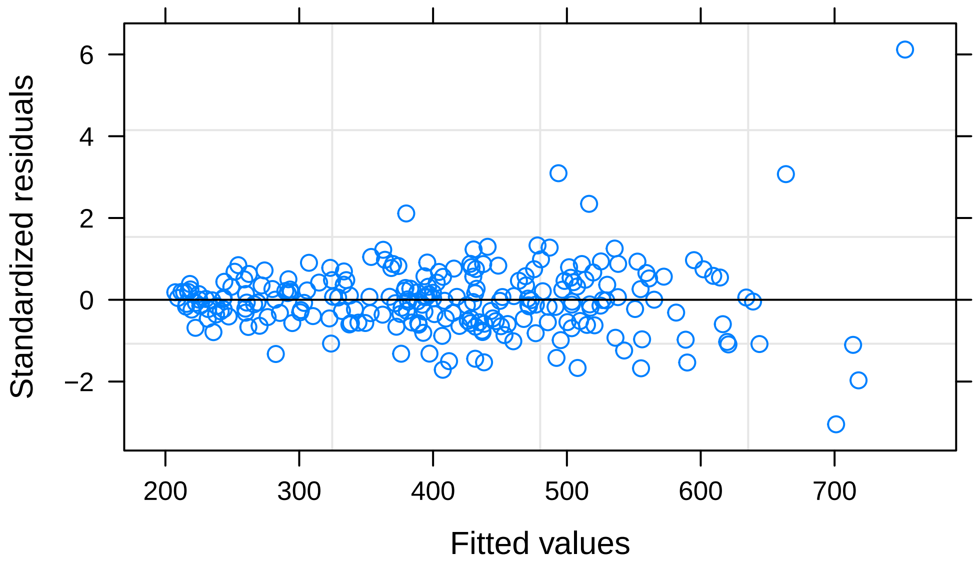
Update the model:

```
m2 <- update(m1, cor = corstr)
```

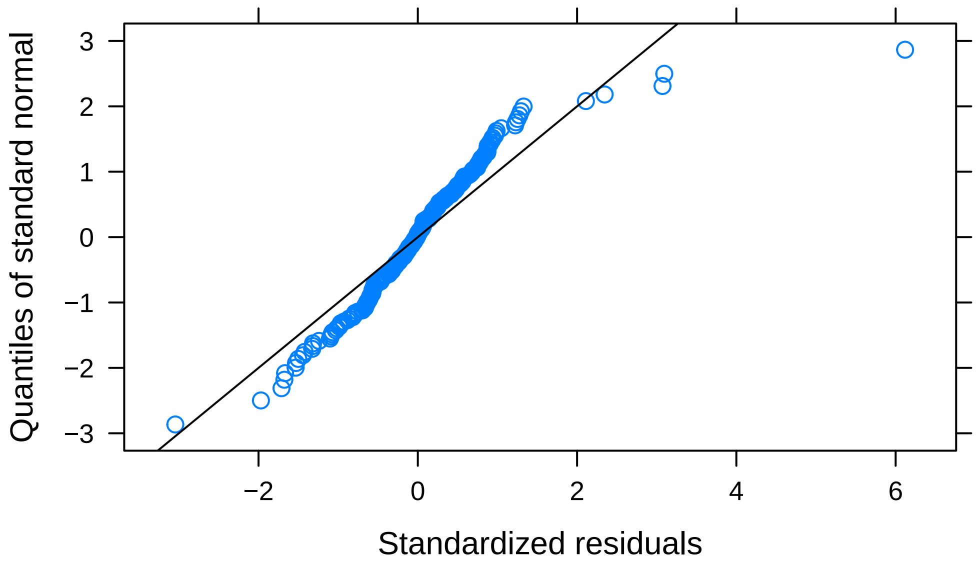
The usual next steps:

check diagnostics

```
plot(m2)
```



```r
qqnorm(m2, ~ resid(., type = "p"), abline = c(0, 1))
```



Look at the variance components.

```r
VarCorr(m2)
```

```
             Variance     StdDev
rep =        pdLogChol(1)
(Intercept)     83.17553    9.120062
irrigation = pdLogChol(1)
(Intercept)    280.54819   16.749573
plot =       pdLogChol(1)
(Intercept)    481.44292   21.941808
Residual     16182.24438  127.209451
```

Run ANOVA

```r
anova(m2)
```

```
                   numDF denDF   F-value p-value
(Intercept)            1   102 1432.6369  <.0001
mix                    9   102   13.6932  <.0001
irrigation             1     3    4.8770  0.1143
cut                    2   102    6.0434  0.0033
mix:irrigation         9   102    0.5256  0.8530
mix:cut               18   102    0.8029  0.6927
irrigation:cut         2   102   14.2649  <.0001
mix:irrigation:cut    18   102    1.0226  0.4418
```

always check the degrees of freedom (denominator and numerator)!

5 Summary

In summary, mixed models are complicated.

References