

Contrasting between- and within-rows stem profiles in poplar plantations - Modelling tree and ‘Treat’ effect - HLM approach

Puletti N., Scotti R., ???

june 2018

Contents

Access to the data	1
Level 1 - ‘TreeId’	1
The two orthogonal profiles are significantly different: 1.8 cm +/- 10%	4
Level-2 residuals	14
Opzione non esplorata:	16

Access to the data

```
library(tidyverse)
library(magrittr)
library(broom)

source("DataWrangling.R")
```

*** following: <https://www.jstatsoft.org/article/view/v056i05/> ***

Level 1 - ‘TreeId’

We fit an initial two-level HLM using crossection height (Sect_height) to explain differences between profiles (delta_d) allowing for a random intercept for each tree:

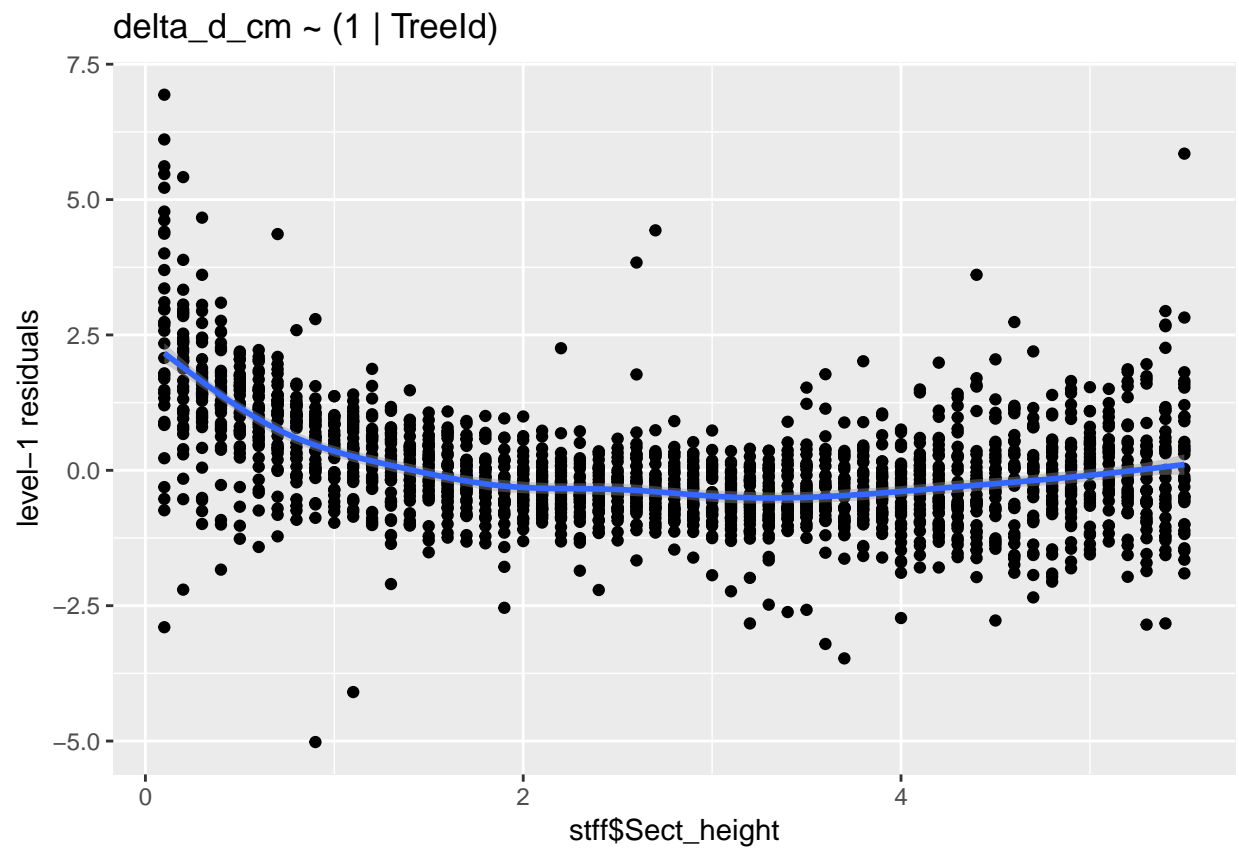
*** delta_d ~ Sect_height + (1 | TreeId) ***

```
library(lme4)

mc <- function(frml, ...) {
  # evaluate the model (specified as formula) and produce the residulas control plot
  fm <- stff %>% lmer(data = ., frml, ... )
  p <- qplot(x = stff$Sect_height, y = residuals(fm),
    geom = c("point", "smooth")) + ylab("level-1 residuals") +
    ggtitle(paste0(as.expression(frml)))
  print(p)
  return(fm)
}
```

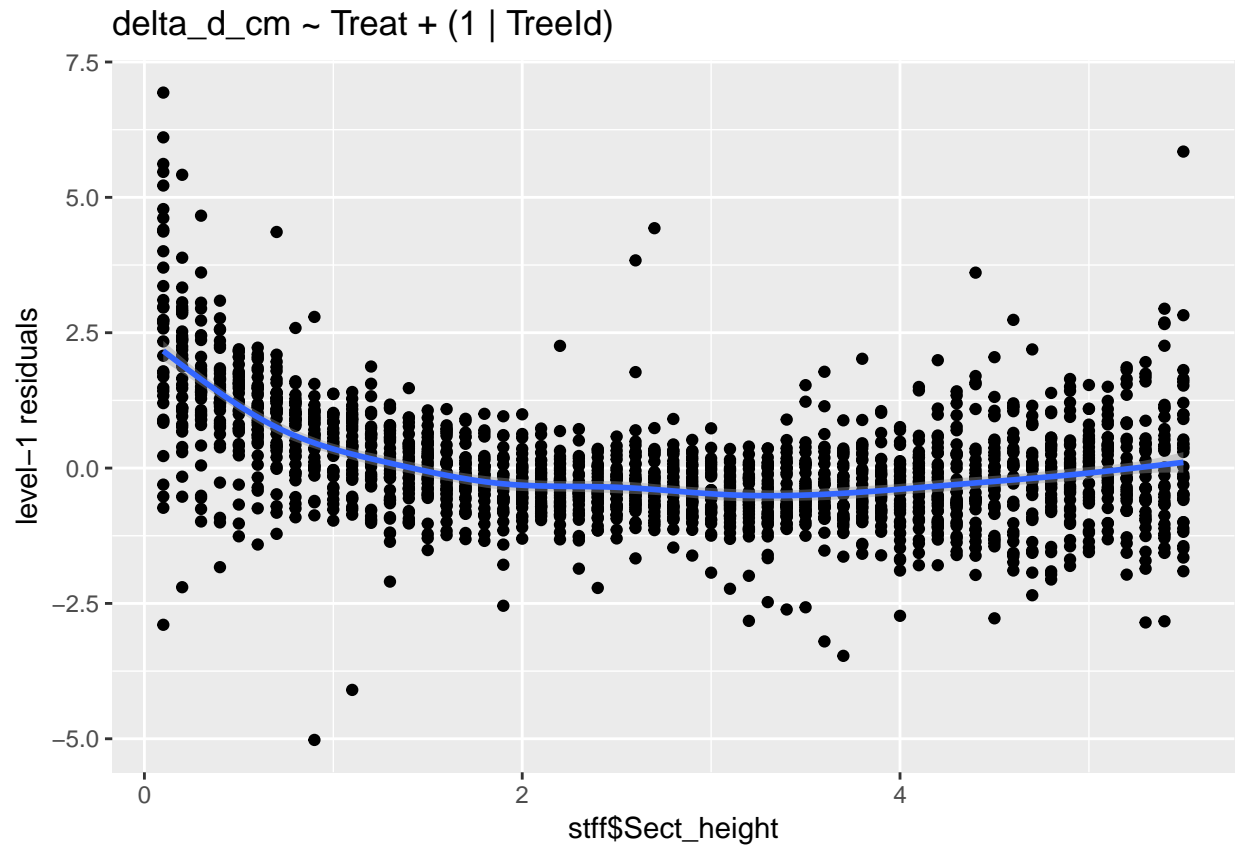
```
# prova senza Treat e senza Sect_height
fm00 <- mc(delta_d_cm ~ (1 | TreeId))
```

```
## `geom_smooth()`` using method = 'gam'
```



```
# prova con Treat ma senza Sect_height
fm10 <- mc(delta_d_cm ~ Treat + (1 | TreeId))
```

```
## `geom_smooth()`` using method = 'gam'
```



```
summary(fm10)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: delta_d_cm ~ Treat + (1 | TreeId)
## Data: .
##
## REML criterion at convergence: 5942
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -4.7039 -0.6199 -0.1528  0.4410  6.4969
##
## Random effects:
##  Groups   Name      Variance Std.Dev.
##  TreeId   (Intercept) 1.209    1.100
##  Residual              1.139    1.067
## Number of obs: 1953, groups: TreeId, 36
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  1.87466    0.32015   5.855
## TreatV450    0.05661    0.45277   0.125
## TreatV500   -0.23917    0.45287  -0.528
##
## Correlation of Fixed Effects:
##              (Intr) TrV450
```

```
## TreatV450 -0.707
## TreatV500 -0.707 0.500

# Treat non ha effetto! (più evidente dopo, con anova)
summary(fm00)

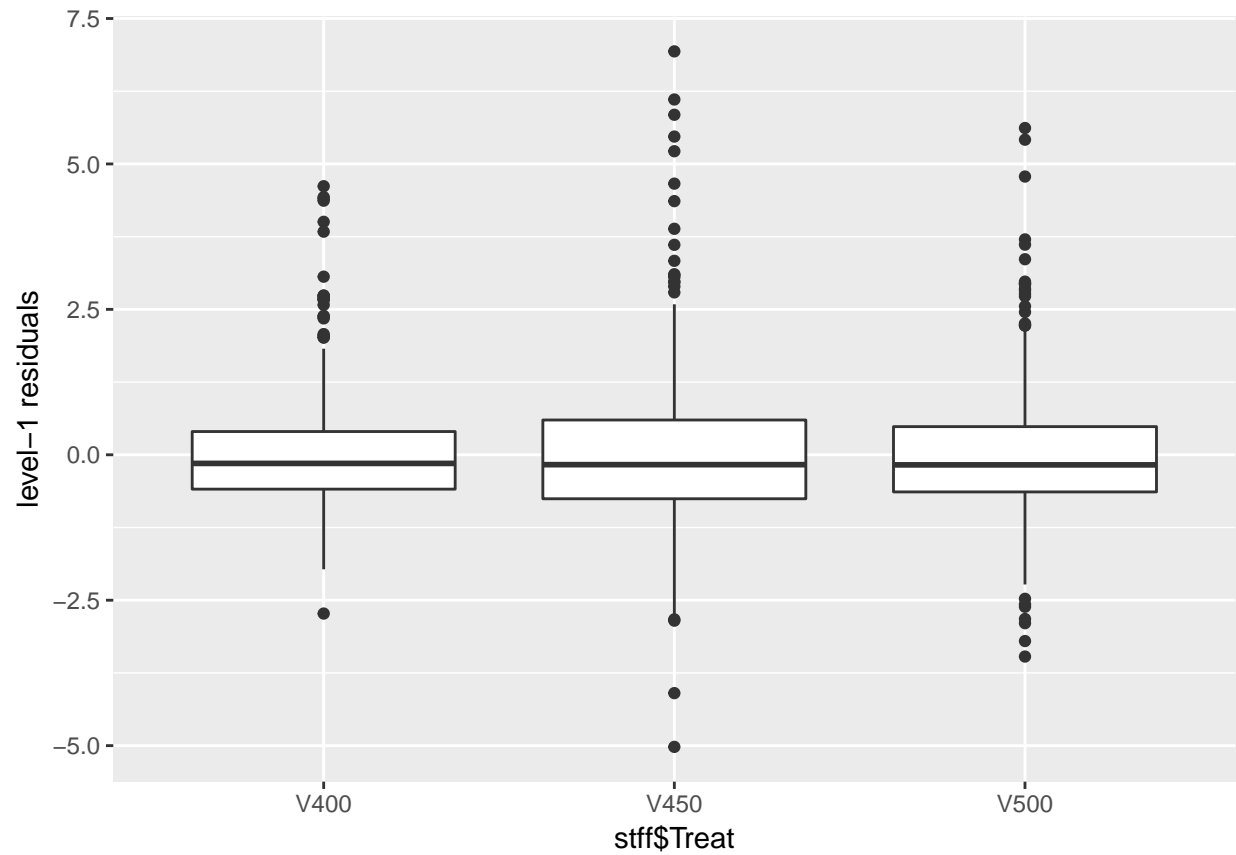
## Linear mixed model fit by REML ['lmerMod']
## Formula: delta_d_cm ~ (1 | TreeId)
## Data: .
##
## REML criterion at convergence: 5942.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -4.7012 -0.6202 -0.1523  0.4425  6.4984
##
## Random effects:
## Groups Name Variance Std.Dev.
## TreeId (Intercept) 1.156 1.075
## Residual 1.139 1.067
## Number of obs: 1953, groups: TreeId, 36
##
## Fixed effects:
## Estimate Std. Error t value
## (Intercept) 1.8139 0.1808 10.03
```

The two orthogonal profiles are significantly different: 1.8 cm +/- 10%

```
anova(fm10, fm00)

## refitting model(s) with ML (instead of REML)
## Data: .
## Models:
## fm00: delta_d_cm ~ (1 | TreeId)
## fm10: delta_d_cm ~ Treat + (1 | TreeId)
##      Df    AIC    BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## fm00  3 5947.1 5963.8 -2970.6 5941.1
## fm10  5 5950.6 5978.5 -2970.3 5940.6 0.5205 2 0.7708

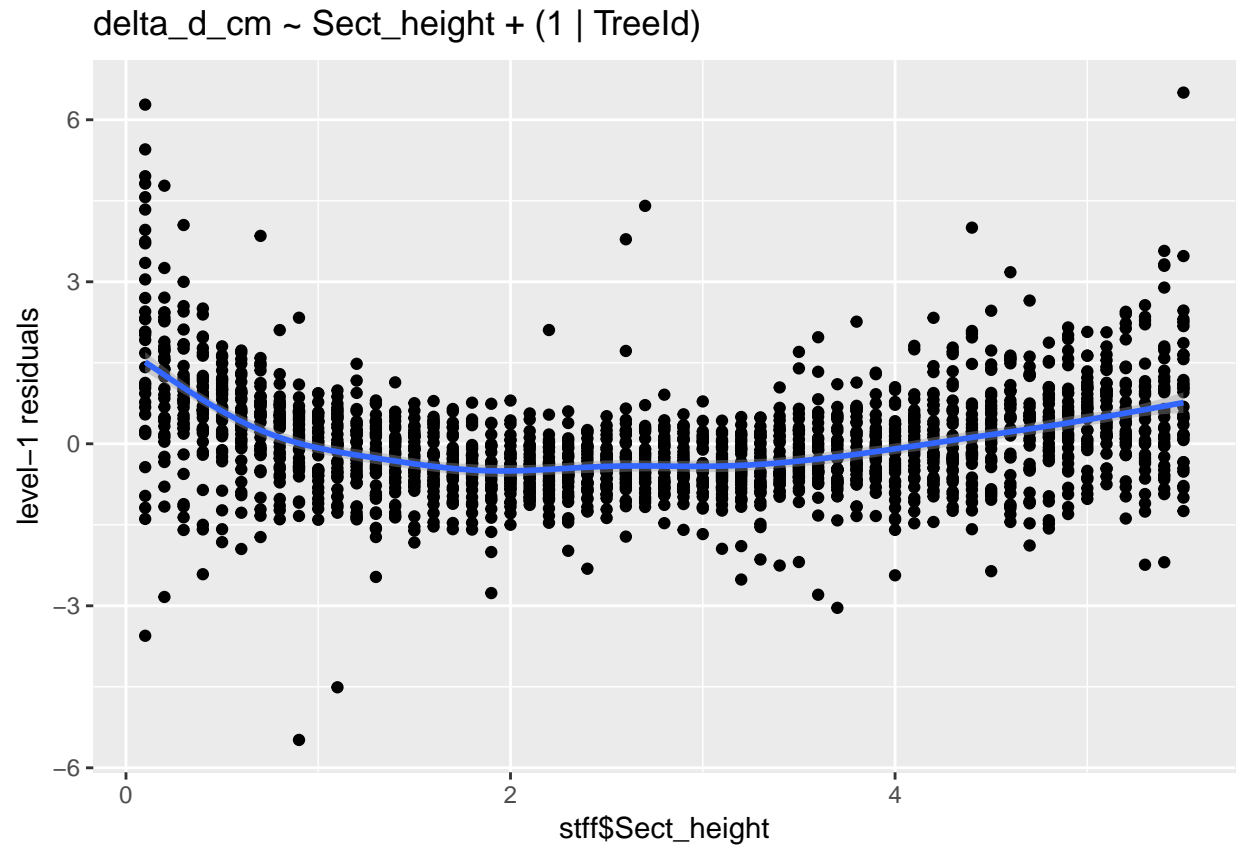
# Contributo di Treat non significativo
qplot(x = stff$Treat, y = residuals(fm10),
      geom = c("boxplot")) + ylab("level-1 residuals")
```



```
# residui distribuiti male e non a zero di media per Treat
```

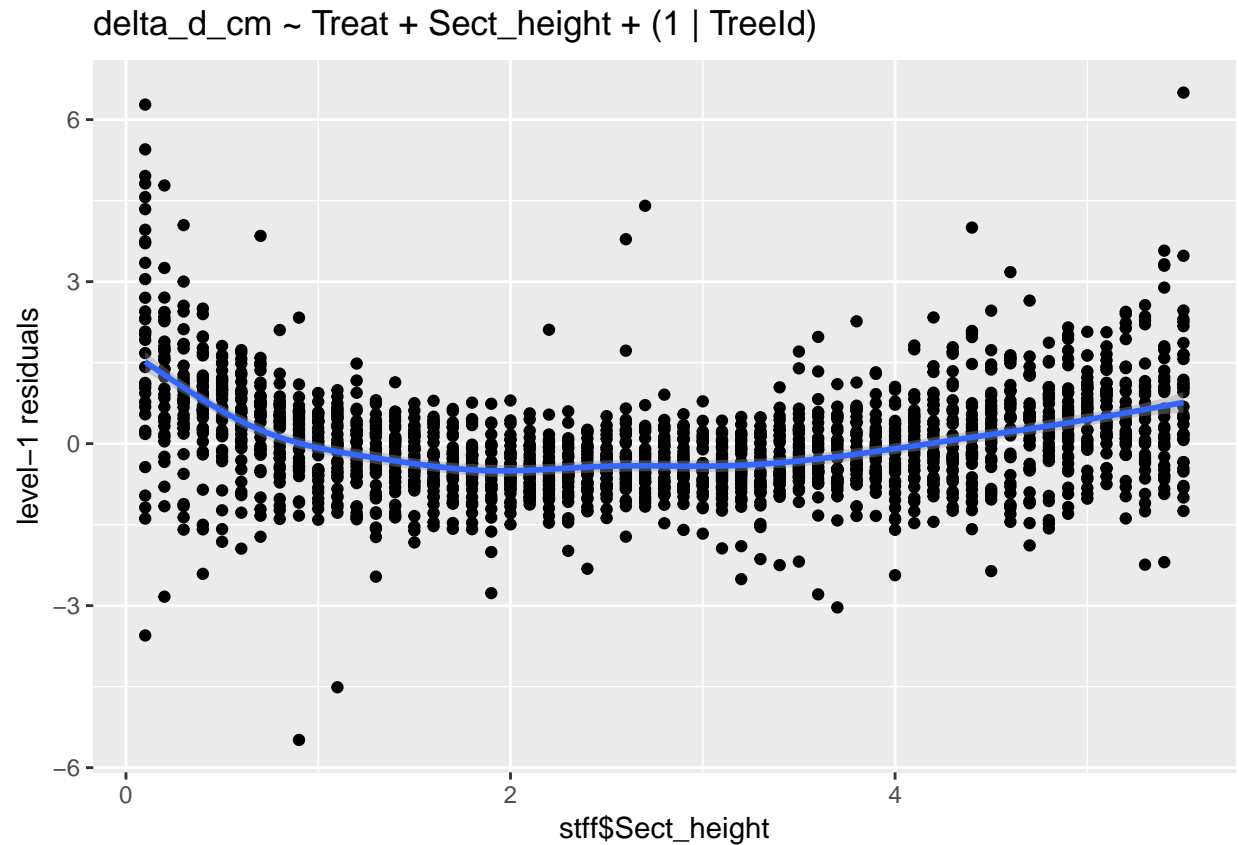
```
# prova CON Sect_height, lineare  
fm01 <- mc(delta_d_cm ~ Sect_height + (1 | TreeId))
```

```
## `geom_smooth()` using method = 'gam'
```



```
fm11 <- mc(delta_d_cm ~ Treat + Sect_height + (1 | TreeId))
```

```
## `geom_smooth()` using method = 'gam'
```



```
anova(fm11, fm01)
```

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

```
## Data: .
```

```
## Models:
```

```
## fm01: delta_d_cm ~ Sect_height + (1 | TreeId)
```

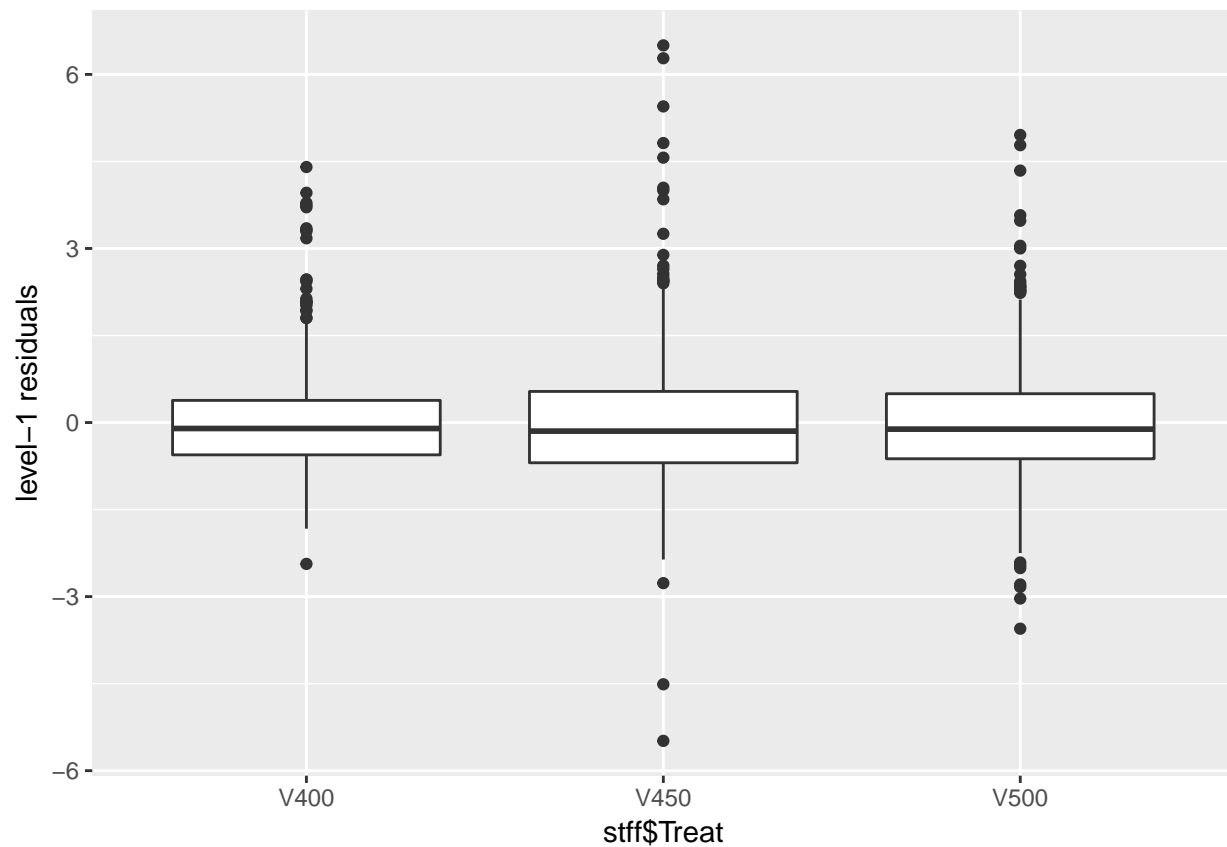
```
## fm11: delta_d_cm ~ Treat + Sect_height + (1 | TreeId)
```

```
##      Df    AIC    BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
```

```
## fm01  4 5679.2 5701.6 -2835.6   5671.2
```

```
## fm11  6 5682.6 5716.1 -2835.3   5670.6 0.6274    2    0.7307
```

```
qplot(x = stff$Treat, y = residuals(fm11),
      geom = c("boxplot")) + ylab("level-1 residuals")
```



```
# residui ancora etreoscedastici
```

```
# prova CON Sect_height, parabolica
```

```
fm12 <- mc(delta_d_cm ~ Treat + (1 | TreeId) + Sect_height *(1 | TreeId) + I(Sect_height^2)*(1 | TreeId))
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
```

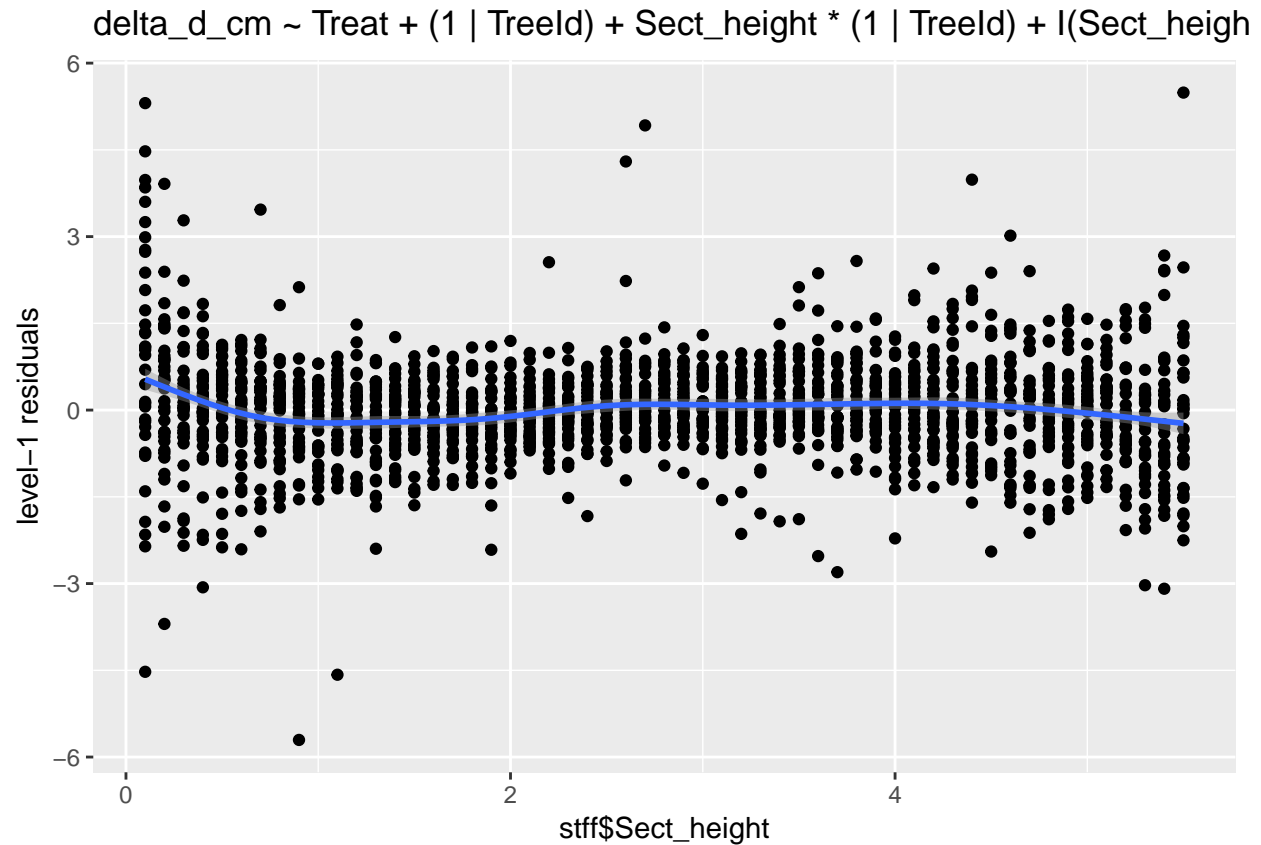
```
## $checkConv, : unable to evaluate scaled gradient
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
```

```
## $checkConv, : Model failed to converge: degenerate Hessian with 1 negative
```

```
## eigenvalues
```

```
## `geom_smooth()` using method = 'gam'
```

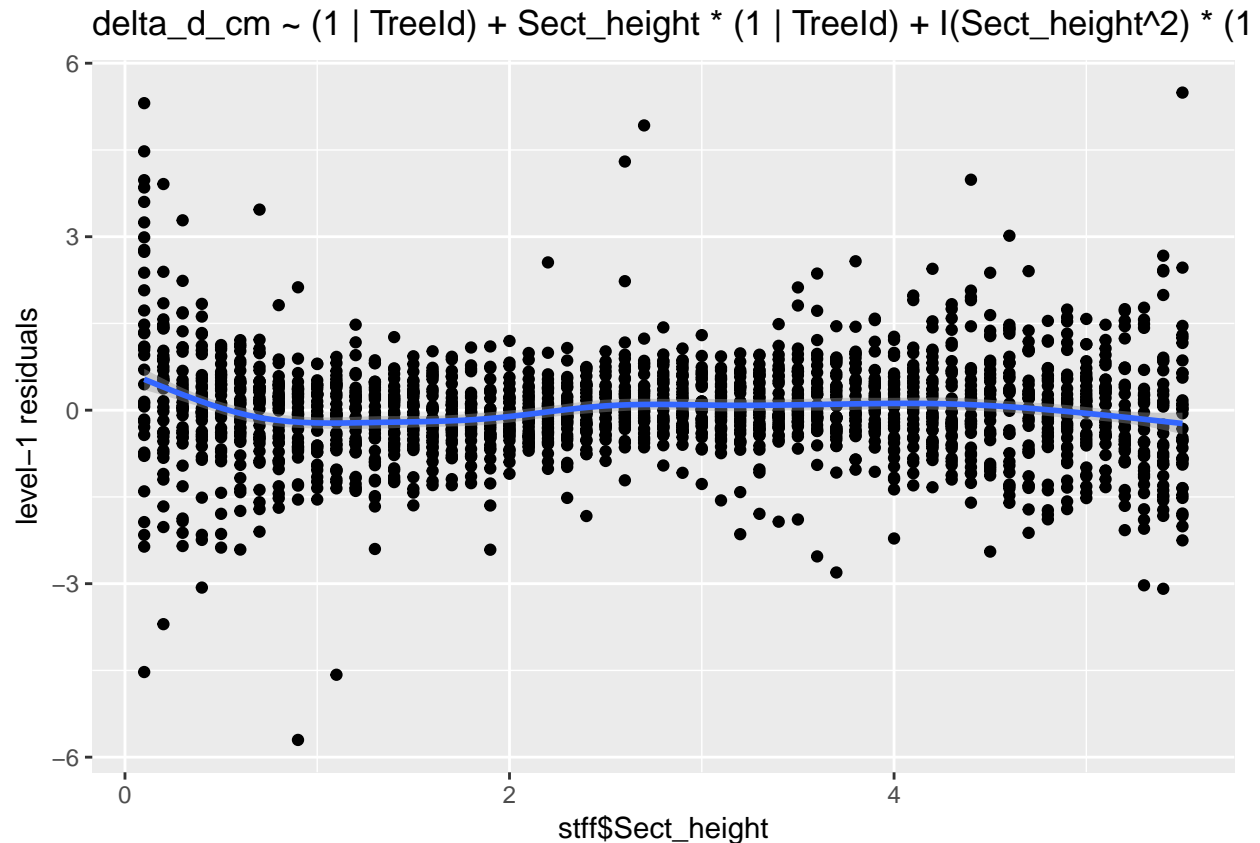



```
fm02 <- mc(delta_d_cm ~ (1 | TreeId) + Sect_height *(1 | TreeId) + I(Sect_height^2)* (1 | TreeId))
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
## $checkConv, : unable to evaluate scaled gradient

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
## $checkConv, : Model failed to converge: degenerate Hessian with 2 negative
## eigenvalues

## `geom_smooth()` using method = 'gam'
```



```
anova(fm02, fm12)
```

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

```
## Data: .
```

```
## Models:
```

```
## fm02: delta_d_cm ~ (1 | TreeId) + Sect_height * (1 | TreeId) + I(Sect_height^2) *
```

```
## fm02:      (1 | TreeId)
```

```
## fm12: delta_d_cm ~ Treat + (1 | TreeId) + Sect_height * (1 | TreeId) +
```

```
## fm12:      I(Sect_height^2) * (1 | TreeId)
```

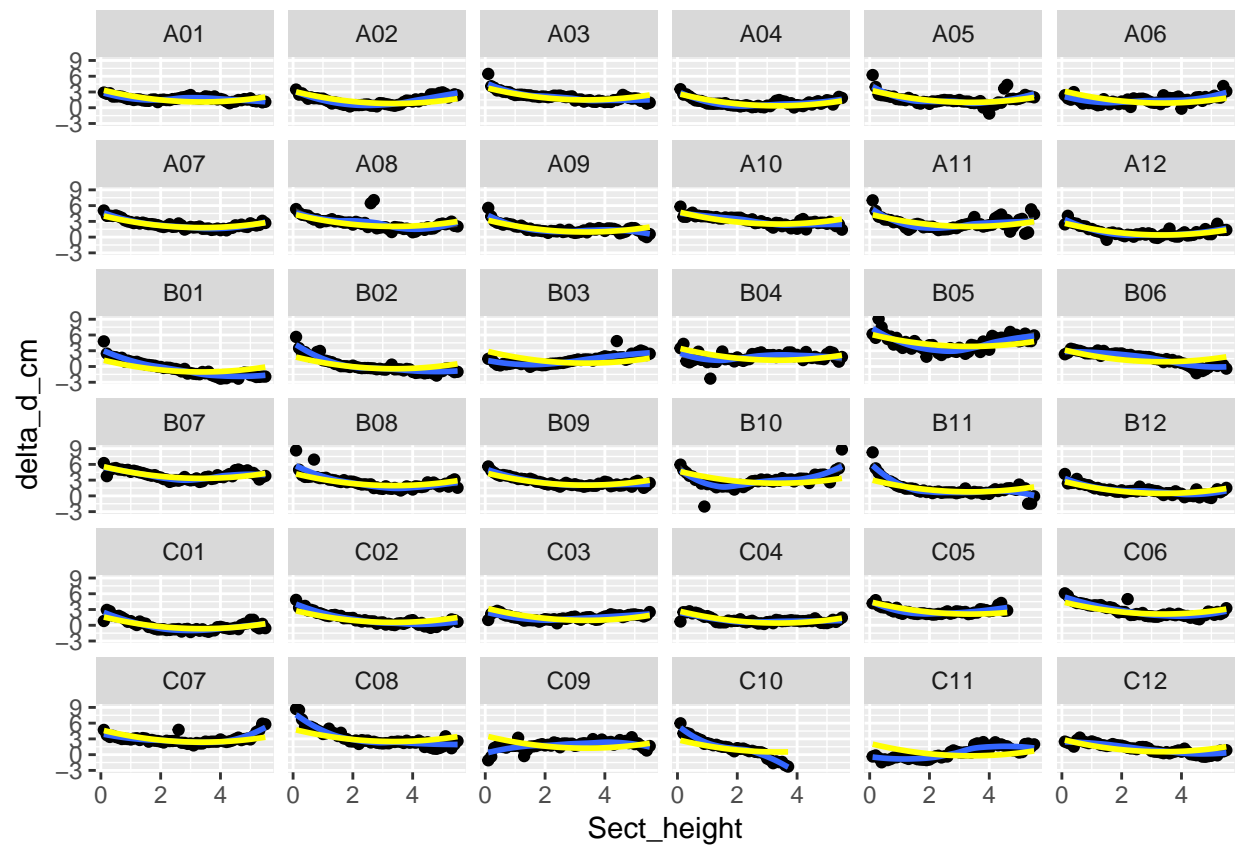
```
##      Df    AIC    BIC logLik deviance Chisq Chi Df Pr(>Chisq)
```

```
## fm02  7 5200.8 5239.8 -2593.4  5186.8
```

```
## fm12  9 5204.2 5254.4 -2593.1  5186.2 0.5464      2    0.7609
```

```
stff %>% ggplot() +
  geom_point(aes(x = Sect_height, y = delta_d_cm)) +
  geom_smooth(aes(x = Sect_height, y = delta_d_cm)) +
  geom_line(aes(x = Sect_height, y = fitted(fm12)), color = "yellow", size = 1) +
  facet_wrap(~TreeId)
```

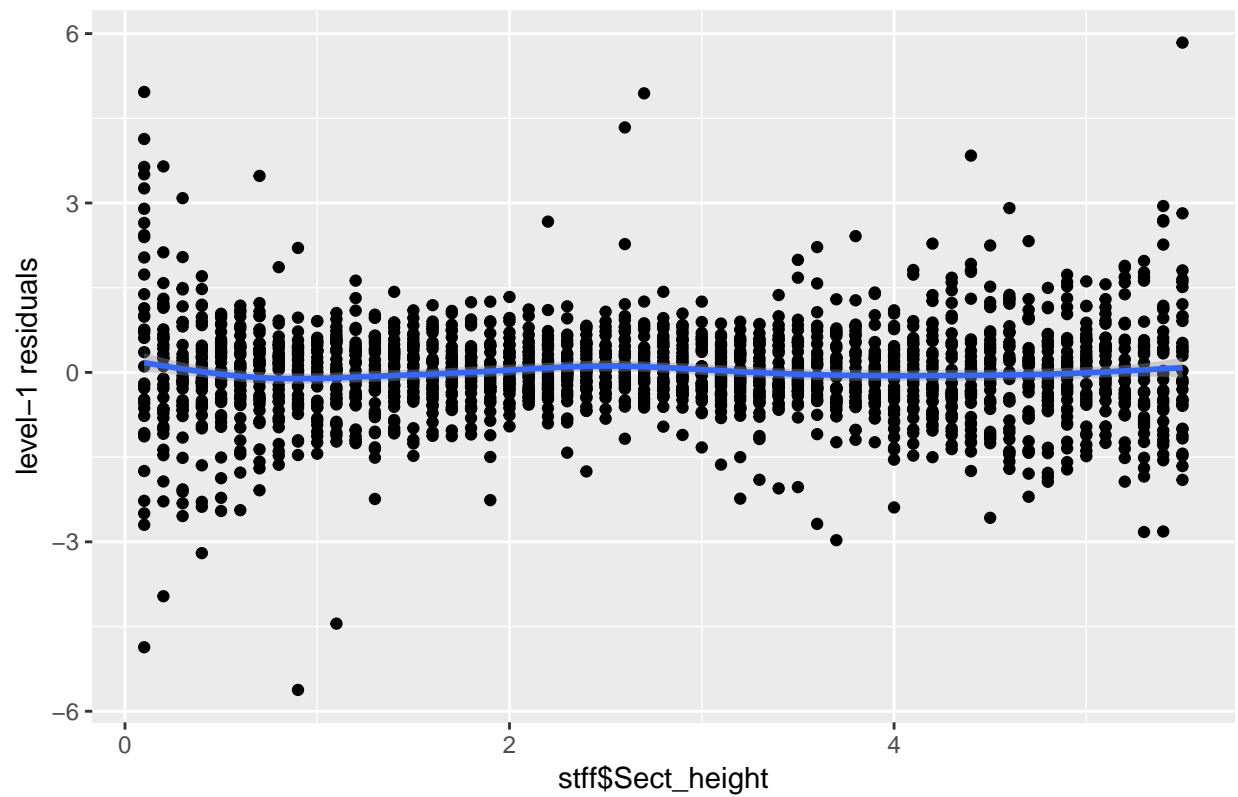
```
## `geom_smooth()` using method = 'loess'
```



```
# prova CON Sect_height, al cubo (con RML=T, non converge!)
fm13 <- mc(delta_d_cm ~ Treat + (1 | TreeId)
  + Sect_height *(1 | TreeId)
  + I(Sect_height^2)* (1 | TreeId)
  + I(Sect_height^3)* (1 | TreeId)
  , REML = F)
```

```
## `geom_smooth()` using method = 'gam'
```

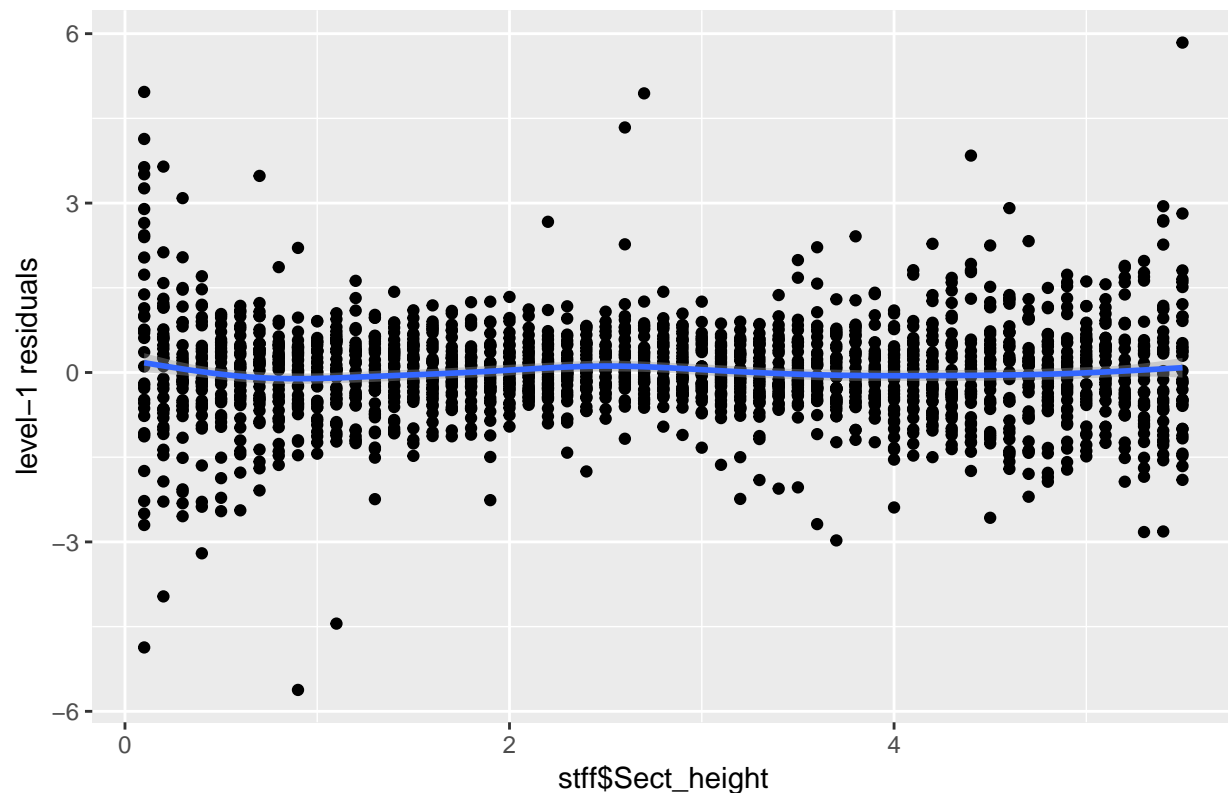
delta_d_cm ~ Treat + (1 | TreeId) + Sect_height * (1 | TreeId) + I(Sect_heigh



```
fm03 <- mc(delta_d_cm ~ (1 | TreeId)
  + Sect_height *(1 | TreeId)
  + I(Sect_height^2)* (1 | TreeId)
  + I(Sect_height^3)* (1 | TreeId)
  , REML = F)
```

```
## `geom_smooth()` using method = 'gam'
```

$\text{delta_d_cm} \sim (1 \mid \text{TreeId}) + \text{Sect_height} * (1 \mid \text{TreeId}) + \text{I}(\text{Sect_height}^2) * (1$



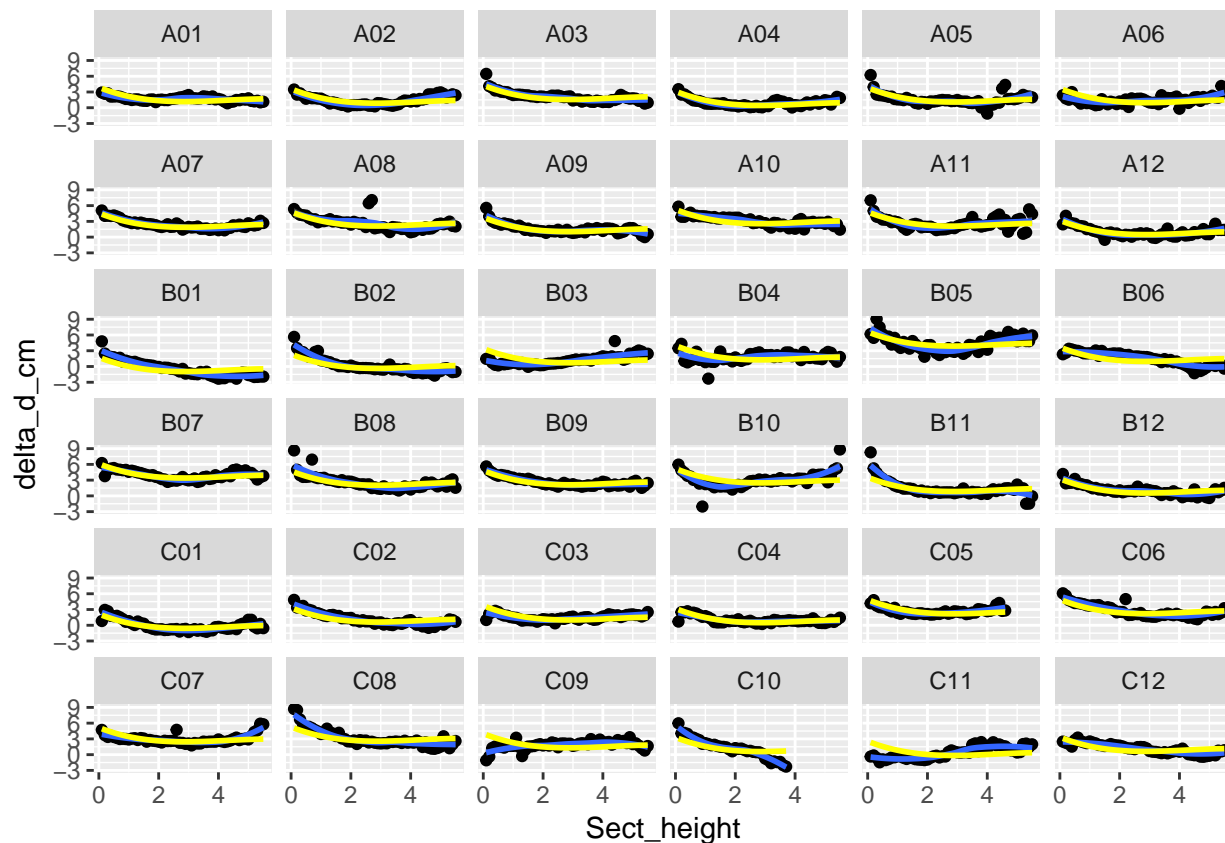
```
anova(fm03, fm13)
```

```
## Data: .
## Models:
## fm03: delta_d_cm ~ (1 | TreeId) + Sect_height * (1 | TreeId) + I(Sect_height^2) *
## fm03:      (1 | TreeId) + I(Sect_height^3) * (1 | TreeId)
## fm13: delta_d_cm ~ Treat + (1 | TreeId) + Sect_height * (1 | TreeId) +
## fm13:      I(Sect_height^2) * (1 | TreeId) + I(Sect_height^3) * (1 |
## fm13:      TreeId)
##      Df    AIC    BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## fm03  9 5151.0 5201.2 -2566.5   5133.0
## fm13 11 5154.4 5215.8 -2566.2   5132.4 0.5501    2    0.7595
```

```
# l'effetto di Treat, non si manifesta in nessun modo!!
# Pr(>Chisq) = 0.761 !! Conferma, non si vede differenza
```

```
stff %>% ggplot() +
  geom_point(aes(x = Sect_height, y = delta_d_cm)) +
  geom_smooth(aes(x = Sect_height, y = delta_d_cm)) +
  geom_line(aes(x = Sect_height, y = fitted(fm13)), color = "yellow", size = 1) +
  facet_wrap(~TreeId)
```

```
## `geom_smooth()` using method = 'loess'
```



In an exploratory setting we would go through this cycle of residual analysis, identification of explanatory variables, and model fitting multiple times until a satisfactory level-1 model is found (Tukey 1977).

Level-2 residuals

The level-2, or random effects, residuals are defined as Zibi or, more commonly, bi. Following an upward residual analysis we prefer the use of EB residuals at level 2 (instead of LS residuals). The level-2 residuals are used to * identify additional explanatory variables that contribute significantly to the model, * check linearity of the level-2 explanatory variables, and * investigate whether the level-2 residuals follow a normal distribution. To obtain the level-2 EB residuals from model fm3, we use the following code:

```
resid2_fm13 <- HLMresid(object = fm13, level = "TreeId")
head(resid2_fm13)
```

```
##      (Intercept) (Intercept) (Intercept) (Intercept)
## A01 -0.005603814 -0.05534186 -0.05495312 -0.06592019
## A02 -0.014895308 -0.14710232 -0.14606902 -0.17522022
## A03  0.006472235  0.06391817  0.06346918  0.07613581
## A04 -0.028426347 -0.28073146 -0.27875951 -0.33439193
## A05 -0.008099787 -0.07999147 -0.07942958 -0.09528145
## A06 -0.012045868 -0.11896197 -0.11812634 -0.14170097
```

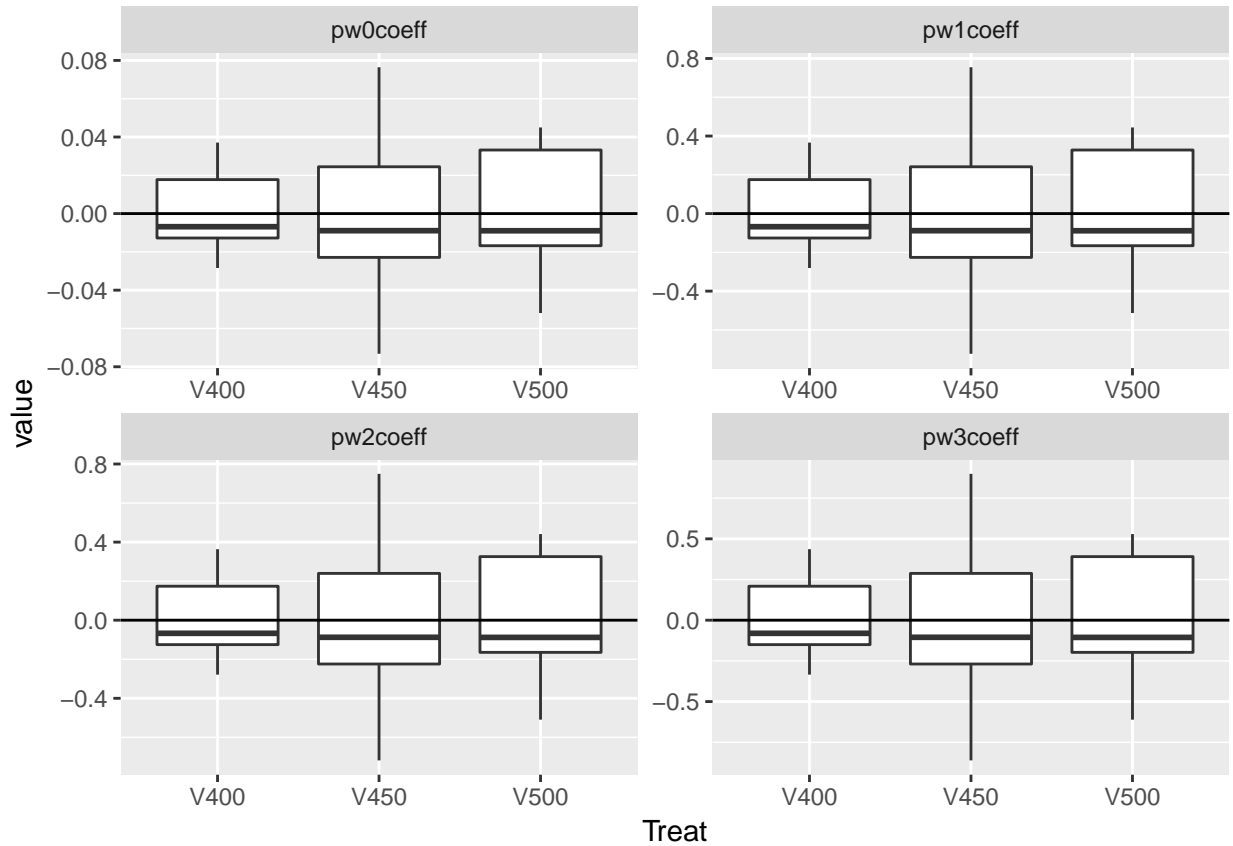
```
res <- stiff %>%
  select(Treat, TreeId) %>%
  unique() %>%
  inner_join(mutate(resid2_fm13, TreeId = row.names(resid2_fm12))) %>%
```

```
select(1:2, pw0coeff = 3, pw1coeff = 4, pw2coeff = 5, pw3coeff = 6) %>%
gather(param, value, 3:6)
```

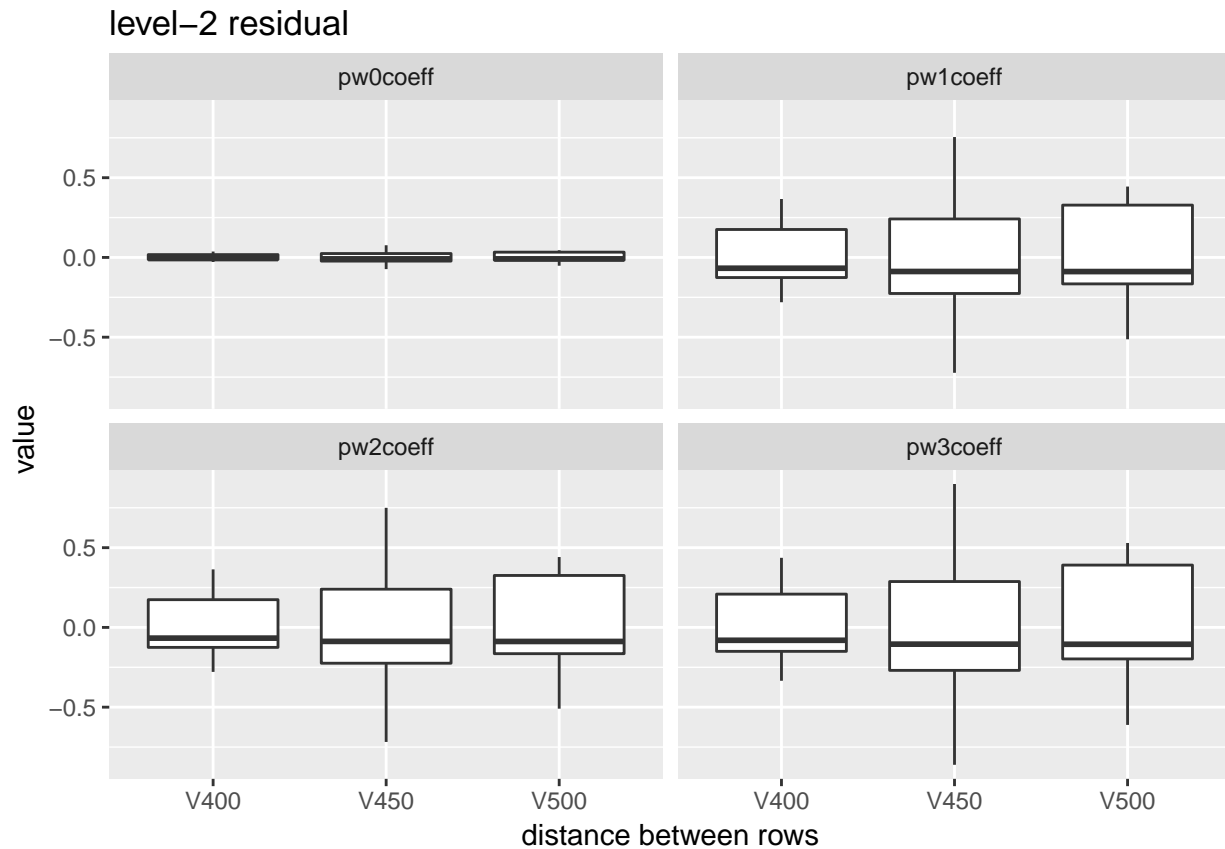
```
## Joining, by = "TreeId"
```

```
# paste0("pw",0:3,"coeff = ", 3:6, collapse=", ")
```

```
ggplot(res, aes(x = Treat, y = value)) +
  geom_boxplot() +
  geom_hline(yintercept = 0) +
  facet_wrap(~param, scales = "free")
```



```
res %$%
  qqplot(x = Treat, y = value, geom = 'boxplot', facets = ~param,
  xlab = 'distance between rows', main = 'level-2 residual')
```



Opzione non esplorata:

Forse è più semplice dividere il range di Sect_height in 3 parti < 1 e > 4 (zeri dell'interpolazione generale dei residui dal modello lineare) ed utilizzare quats avriabile come ulteriore 'effetto fisso'. A quel punto l'approssimazione lineare di $\Delta_d = f(\text{Sect_height})$ dovrebbe essere presentabile.

```
[rmarkdown::render("PdP_HLM3.Rmd", encoding="UTF-8")]
```