Gina Nichols - GAMs

Miranda Tilton

October 30, 2020

Data processing

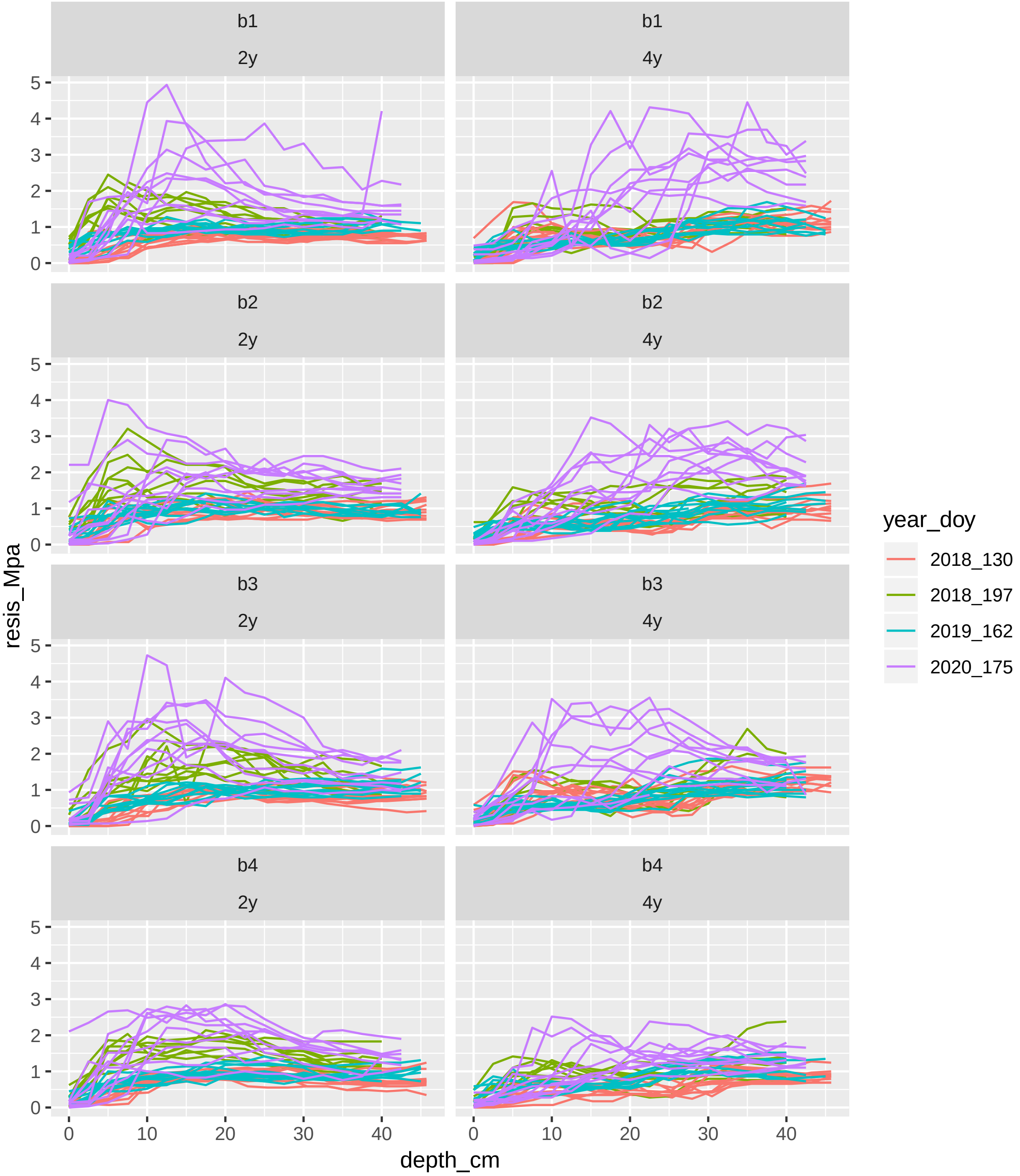
```
# remotes::install_github("vanichols/maRsden")
library(magrittr)
library(maRsden)
library(dplyr)
library(ggplot2)
library(lemon) # chunk option `render = lemon_print` makes tables prettier
```

```
myd <-
  mrs_penetrom %>%
  left_join(mrs_plotkey) %>%
  #filter(year != "2020") %>%
  mutate(resis_Mpa = resis_kpa/1000) %>%
  select(year, doy, block, rot_trt, plot_id, rep_id, depth_cm, resis_Mpa) %>%
  arrange(block, plot_id, rep_id, depth_cm)

# make new factor variables and convert old trt/block variables into factors
# sorry, I couldn't figure out how to do this in mutate() without gnarly warnings
myd$year_doy <- as.factor(paste(myd$year, myd$doy, sep = "_"))
myd$trt_block_yr <- as.factor(paste(myd$rot_trt, myd$block, myd$year, myd$doy, sep = "_"))
myd$block <- as.factor(myd$block)
myd$rot_trt <- as.factor(myd$rot_trt)

head(myd, 12)
```

year	doy	block	rot_trt	plot_id	rep_id	depth_cm	resis_Mpa	year_doy	trt_block_yr
2018	130	b1	2y	2018_13	2018_13-1	0.00	0.0000000	2018_130	2y_b1_2018_130
2018	130	b1	2y	2018_13	2018_13-1	2.54	0.8618450	2018_130	2y_b1_2018_130
2018	130	b1	2y	2018_13	2018_13-1	5.08	0.9307926	2018_130	2y_b1_2018_130
2018	130	b1	2y	2018_13	2018_13-1	7.62	0.4481594	2018_130	2y_b1_2018_130
2018	130	b1	2y	2018_13	2018_13-1	10.16	0.6205284	2018_130	2y_b1_2018_130
2018	130	b1	2y	2018_13	2018_13-1	12.70	0.6894760	2018_130	2y_b1_2018_130
2018	130	b1	2y	2018_13	2018_13-1	15.24	0.7584236	2018_130	2y_b1_2018_130
2018	130	b1	2y	2018_13	2018_13-1	17.78	0.7928974	2018_130	2y_b1_2018_130
2018	130	b1	2y	2018_13	2018_13-1	20.32	0.7928974	2018_130	2y_b1_2018_130
2018	130	b1	2y	2018_13	2018_13-1	22.86	0.7584236	2018_130	2y_b1_2018_130
2018	130	b1	2y	2018_13	2018_13-1	25.40	0.6205284	2018_130	2y_b1_2018_130
2018	130	b1	2y	2018_13	2018_13-1	27.94	0.6205284	2018_130	2y_b1_2018_130

Data visualization

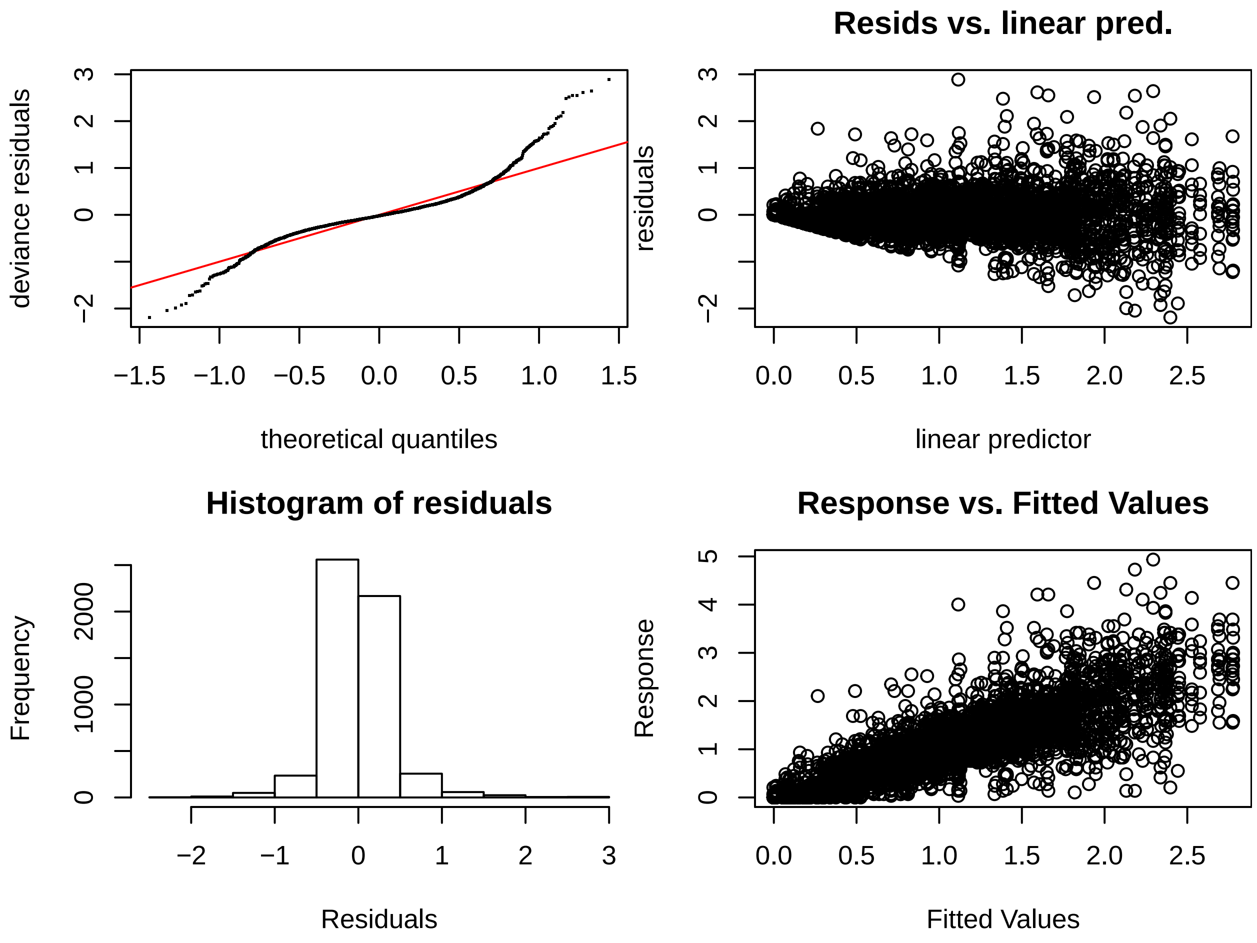
```
# may want to account for interaction w/ year? looks like pattern changes over time
ggplot(data = myd) +
  geom_line(aes(x = depth_cm, y = resis_Mpa, group = rep_id, color = year_doy)) +
  facet_wrap( ~ block + rot_trt, ncol = 2)
```



Fit and check GAM

```r
library(mgcv)
mod <- gam(resis_Mpa ~ s(depth_cm, by = trt_block_yr, bs = "cr", k = 8) + trt_block_yr,
           data = myd, method = "REML")
```

```r
# plot(mod, residuals = TRUE, shade = TRUE)
par(mar = c(4, 4, 3, 0))
mgcv::gam.check(mod)
```



```
##
## Method: REML   Optimizer: outer newton
## full convergence after 10 iterations.
## Gradient range [-0.0009455196,0.0008441727]
## (score 2710.324 & scale 0.1481622).
## Hessian positive definite, eigenvalue range [3.297462e-05,2653.529].
## Model rank =  256 / 256
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
```

```
##
##                                          k'  edf k-index p-value
## s(depth_cm):trt_block_yr2y_b1_2018_130 7.00 3.25    0.98   0.095 .
## s(depth_cm):trt_block_yr2y_b1_2018_197 7.00 4.98    0.98   0.145
## s(depth_cm):trt_block_yr2y_b1_2019_162 7.00 3.22    0.98   0.130
## s(depth_cm):trt_block_yr2y_b1_2020_175 7.00 5.98    0.98   0.140
## s(depth_cm):trt_block_yr2y_b2_2018_130 7.00 3.73    0.98   0.135
## s(depth_cm):trt_block_yr2y_b2_2018_197 7.00 5.36    0.98   0.110
## s(depth_cm):trt_block_yr2y_b2_2019_162 7.00 3.44    0.98   0.130
## s(depth_cm):trt_block_yr2y_b2_2020_175 7.00 4.42    0.98   0.120
## s(depth_cm):trt_block_yr2y_b3_2018_130 7.00 3.63    0.98   0.160
## s(depth_cm):trt_block_yr2y_b3_2018_197 7.00 4.43    0.98   0.115
## s(depth_cm):trt_block_yr2y_b3_2019_162 7.00 3.32    0.98   0.145
## s(depth_cm):trt_block_yr2y_b3_2020_175 7.00 5.66    0.98   0.090 .
## s(depth_cm):trt_block_yr2y_b4_2018_130 7.00 3.57    0.98   0.155
## s(depth_cm):trt_block_yr2y_b4_2018_197 7.00 4.96    0.98   0.075 .
## s(depth_cm):trt_block_yr2y_b4_2019_162 7.00 3.30    0.98   0.090 .
## s(depth_cm):trt_block_yr2y_b4_2020_175 7.00 5.01    0.98   0.165
## s(depth_cm):trt_block_yr4y_b1_2018_130 7.00 4.36    0.98   0.090 .
## s(depth_cm):trt_block_yr4y_b1_2018_197 7.00 4.50    0.98   0.110
## s(depth_cm):trt_block_yr4y_b1_2019_162 7.00 1.00    0.98   0.155
## s(depth_cm):trt_block_yr4y_b1_2020_175 7.00 4.74    0.98   0.105
## s(depth_cm):trt_block_yr4y_b2_2018_130 7.00 3.57    0.98   0.110
## s(depth_cm):trt_block_yr4y_b2_2018_197 7.00 4.23    0.98   0.180
## s(depth_cm):trt_block_yr4y_b2_2019_162 7.00 1.00    0.98   0.120
## s(depth_cm):trt_block_yr4y_b2_2020_175 7.00 4.31    0.98   0.115
## s(depth_cm):trt_block_yr4y_b3_2018_130 7.00 5.22    0.98   0.120
## s(depth_cm):trt_block_yr4y_b3_2018_197 7.00 4.41    0.98   0.110
## s(depth_cm):trt_block_yr4y_b3_2019_162 7.00 1.00    0.98   0.115
## s(depth_cm):trt_block_yr4y_b3_2020_175 7.00 4.46    0.98   0.105
## s(depth_cm):trt_block_yr4y_b4_2018_130 7.00 1.00    0.98   0.110
## s(depth_cm):trt_block_yr4y_b4_2018_197 7.00 4.33    0.98   0.100 .
## s(depth_cm):trt_block_yr4y_b4_2019_162 7.00 3.10    0.98   0.165
## s(depth_cm):trt_block_yr4y_b4_2020_175 7.00 3.75    0.98   0.080 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

To check a GAM with `gam.check()`, we look for a few things:

- If `edf` is too close to `k'`, we may need more knots
- `k-index` is the estimate divided by the residual variance. The further below 1 this is, the more likely it is that there is missed pattern left in the residuals.
- The p-value for the `k-index` is computed by simulation: the residuals are randomly re-shuffled `k.rep` times to obtain the null distribution of the differencing variance estimator, if there is no pattern in the residuals.
- If the p-value is too close to zero, there is a significant pattern in the residuals that should be addressed.

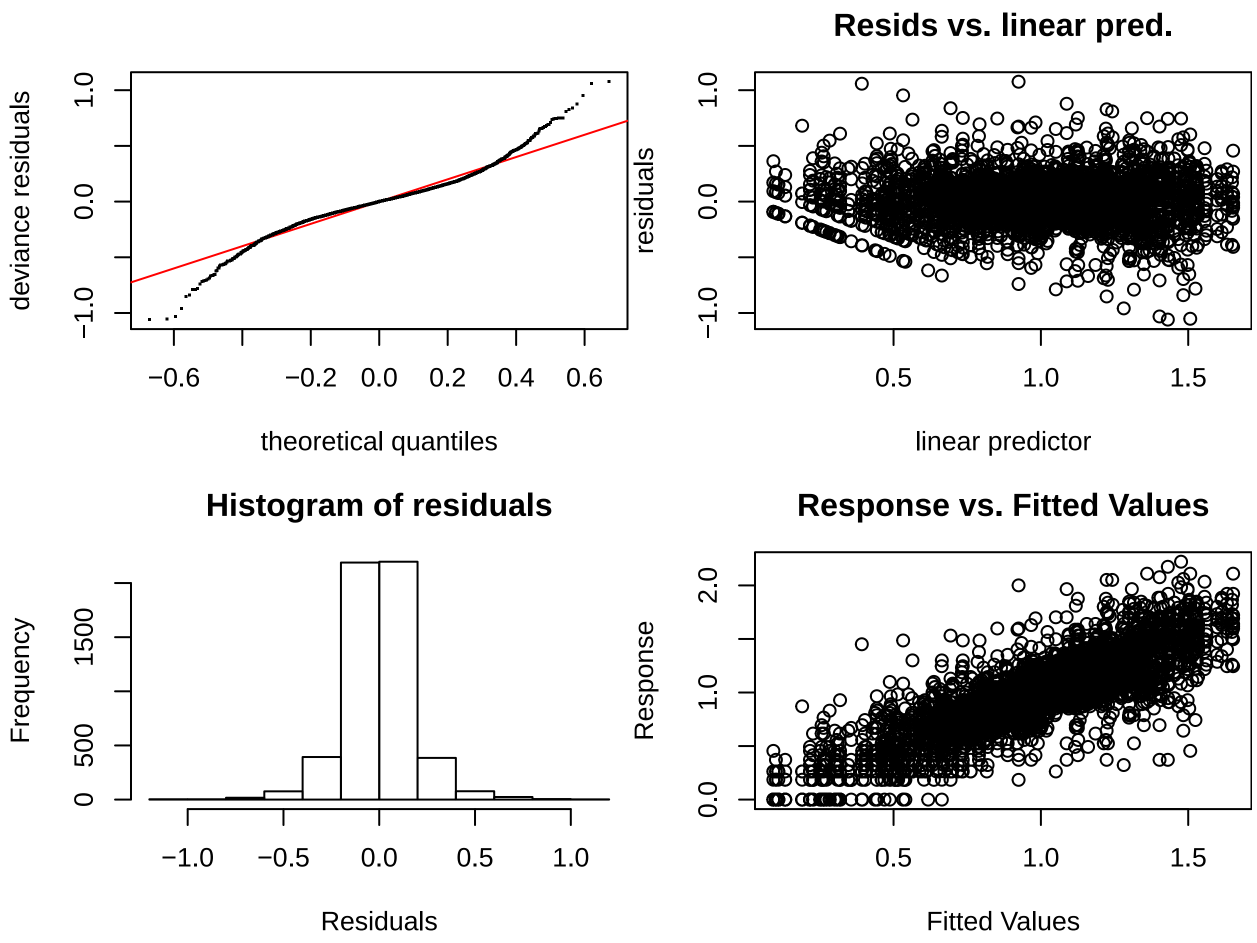
In this case, the `edf` values are not too close to 7, and the `k-index` is close to 1, but the p-values are a bit low for my taste. This is likely due to the heteroskedastic residuals (i.e., megaphone shape), because the curves are all close to zero at zero depth but spread out a lot for higher depths.

Next, I transform `resis_Mpa` with the square root (since all resistance values are non-negative) and fit another model, to unify/control variances at high depths.

Adjust for non-constant variance of `depth_cm` and refit

```
mod <- gam(sqrt(resis_Mpa) ~ s(depth_cm, by = trt_block_yr, bs = "cr", k = 8) + trt_block_yr,
           data = myd, method = "REML")
```

```
# plot(mod, residuals = TRUE, shade = TRUE)
par(mar = c(4, 4, 3, 0))
mgcv::gam.check(mod)
```



```
##
## Method: REML   Optimizer: outer newton
## full convergence after 7 iterations.
## Gradient range [-0.0007632464,0.002795664]
## (score -1269.941 & scale 0.03229665).
## Hessian positive definite, eigenvalue range [0.000776348,2653.545].
## Model rank =  256 / 256
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
```

```
##                                           k'  edf k-index p-value
## s(depth_cm):trt_block_yr2y_b1_2018_130 7.00 4.85       1    0.36
## s(depth_cm):trt_block_yr2y_b1_2018_197 7.00 5.18       1    0.42
## s(depth_cm):trt_block_yr2y_b1_2019_162 7.00 3.97       1    0.30
## s(depth_cm):trt_block_yr2y_b1_2020_175 7.00 5.79       1    0.29
## s(depth_cm):trt_block_yr2y_b2_2018_130 7.00 5.45       1    0.34
## s(depth_cm):trt_block_yr2y_b2_2018_197 7.00 5.55       1    0.39
## s(depth_cm):trt_block_yr2y_b2_2019_162 7.00 4.37       1    0.32
## s(depth_cm):trt_block_yr2y_b2_2020_175 7.00 4.79       1    0.37
## s(depth_cm):trt_block_yr2y_b3_2018_130 7.00 4.85       1    0.41
## s(depth_cm):trt_block_yr2y_b3_2018_197 7.00 5.17       1    0.31
## s(depth_cm):trt_block_yr2y_b3_2019_162 7.00 4.66       1    0.35
## s(depth_cm):trt_block_yr2y_b3_2020_175 7.00 5.67       1    0.36
## s(depth_cm):trt_block_yr2y_b4_2018_130 7.00 4.62       1    0.36
## s(depth_cm):trt_block_yr2y_b4_2018_197 7.00 5.51       1    0.37
## s(depth_cm):trt_block_yr2y_b4_2019_162 7.00 4.14       1    0.31
## s(depth_cm):trt_block_yr2y_b4_2020_175 7.00 5.42       1    0.36
## s(depth_cm):trt_block_yr4y_b1_2018_130 7.00 5.58       1    0.36
## s(depth_cm):trt_block_yr4y_b1_2018_197 7.00 5.44       1    0.38
## s(depth_cm):trt_block_yr4y_b1_2019_162 7.00 4.26       1    0.32
## s(depth_cm):trt_block_yr4y_b1_2020_175 7.00 4.35       1    0.42
## s(depth_cm):trt_block_yr4y_b2_2018_130 7.00 5.50       1    0.38
## s(depth_cm):trt_block_yr4y_b2_2018_197 7.00 4.88       1    0.35
## s(depth_cm):trt_block_yr4y_b2_2019_162 7.00 1.01       1    0.34
## s(depth_cm):trt_block_yr4y_b2_2020_175 7.00 4.59       1    0.41
## s(depth_cm):trt_block_yr4y_b3_2018_130 7.00 5.98       1    0.36
## s(depth_cm):trt_block_yr4y_b3_2018_197 7.00 5.24       1    0.34
## s(depth_cm):trt_block_yr4y_b3_2019_162 7.00 4.95       1    0.32
## s(depth_cm):trt_block_yr4y_b3_2020_175 7.00 4.89       1    0.38
## s(depth_cm):trt_block_yr4y_b4_2018_130 7.00 5.50       1    0.43
## s(depth_cm):trt_block_yr4y_b4_2018_197 7.00 5.20       1    0.36
## s(depth_cm):trt_block_yr4y_b4_2019_162 7.00 4.58       1    0.37
## s(depth_cm):trt_block_yr4y_b4_2020_175 7.00 4.38       1    0.34
```

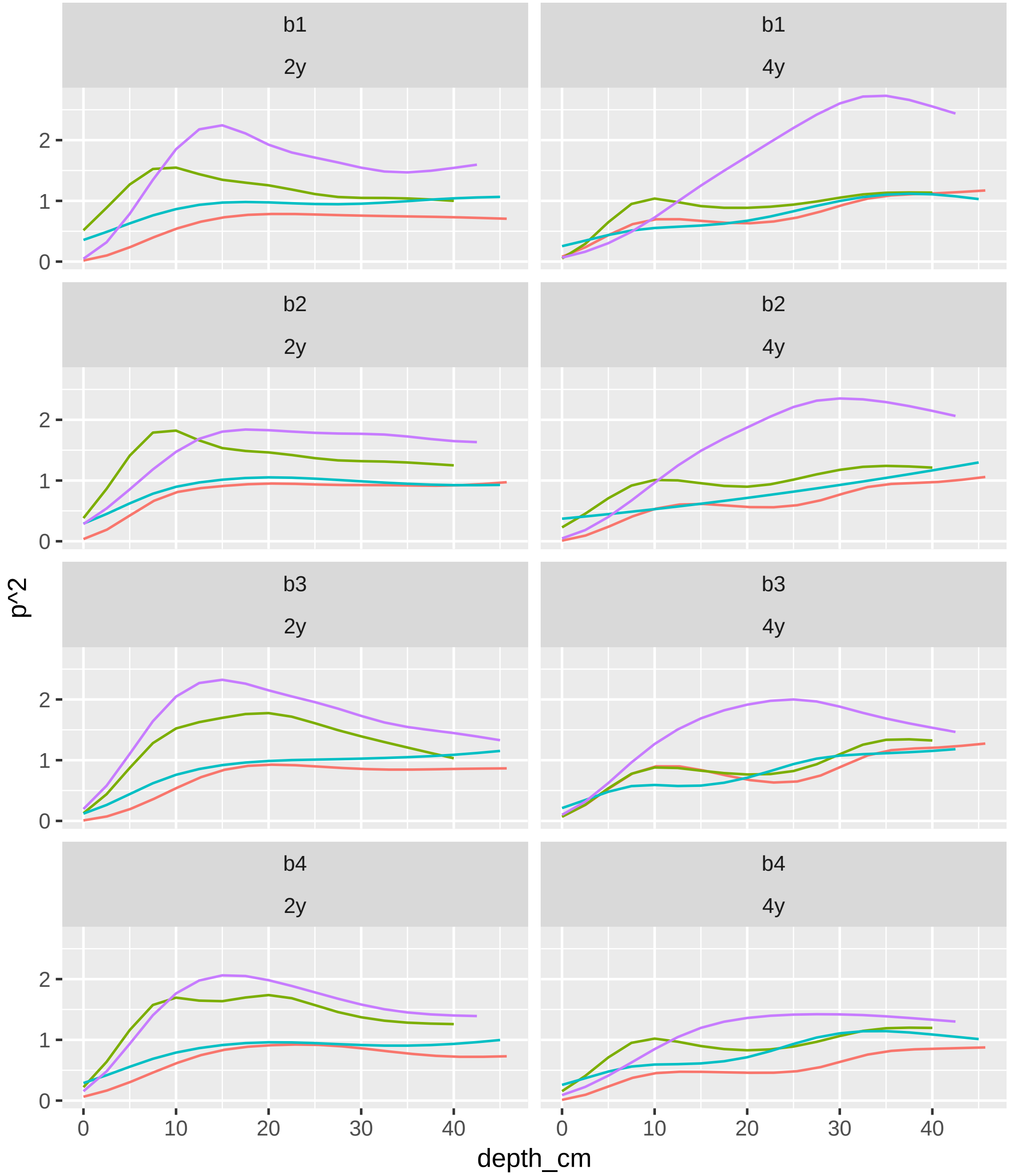
I still don't love that bottomed-out pattern in the residuals (where true resistance is 0 but the model can only be biased positively because all values are non-negative), but the pattern is no longer megaphone-shaped, so this is a good improvement.

The `edf` values are still not too close to `k'`, and the `k-index` values are even closer to 1 than before. Notably, the p-values have gone from $\approx .1$ to $\approx .35$, which I like quite a bit more but still might signal something in the model to be improved upon.

Looking at the residuals plots for each group (two pages further down this document), it seems that there is still an issue with non-constant variance, but this time by year/doy instead of depth. To investigate, I remove this year and fit the model one more time.

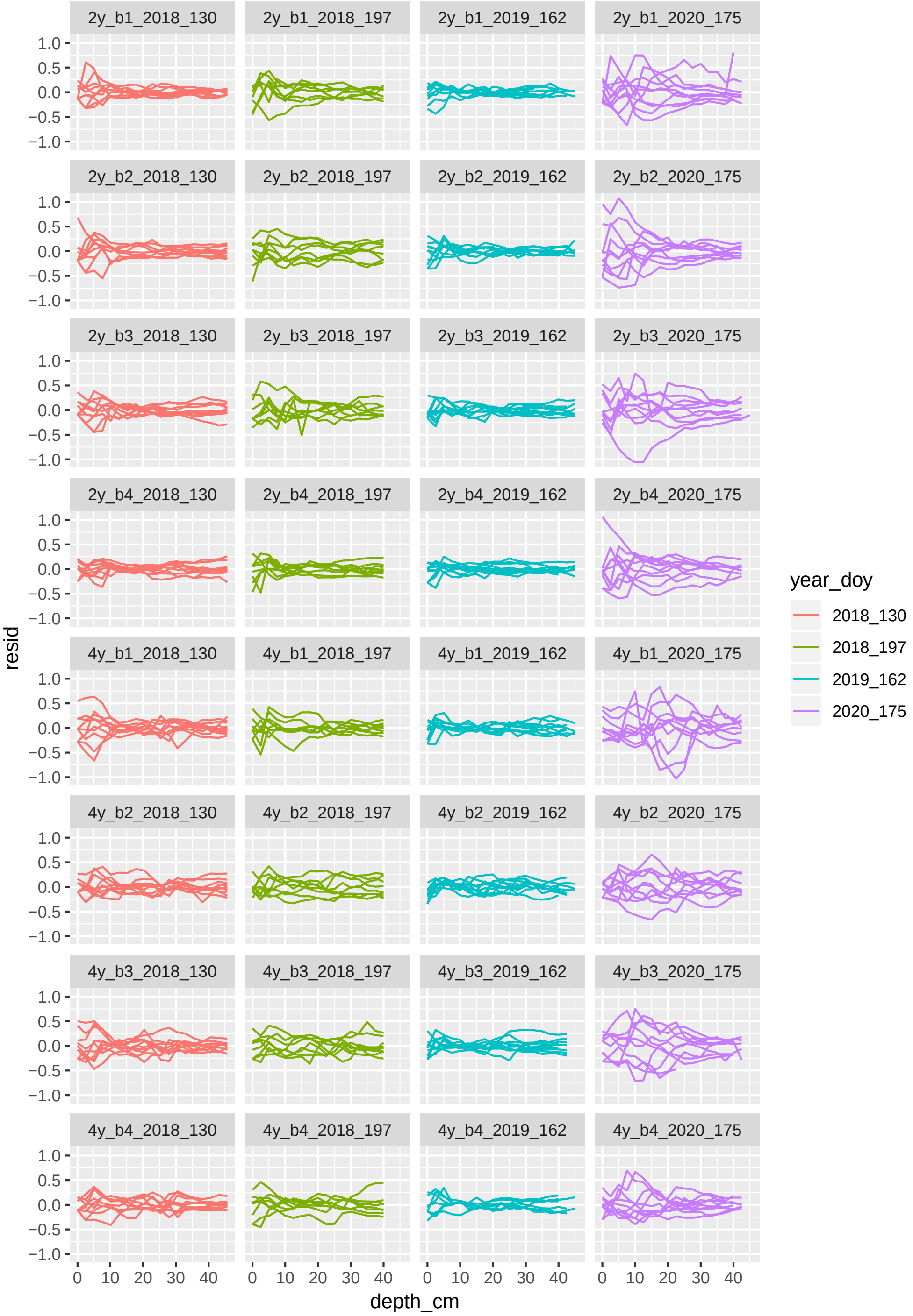
View fitted model by group

```
# view the 24 fitted curves to visually inspect differences
myd$p <- predict(mod)

ggplot(data = myd) +
  geom_line(aes(x = depth_cm, y = p^2, color = year_doy, group = year_doy)) + # note p^2
  facet_wrap( ~ block + rot_trt, ncol = 2) +
  guides(color = FALSE)
```



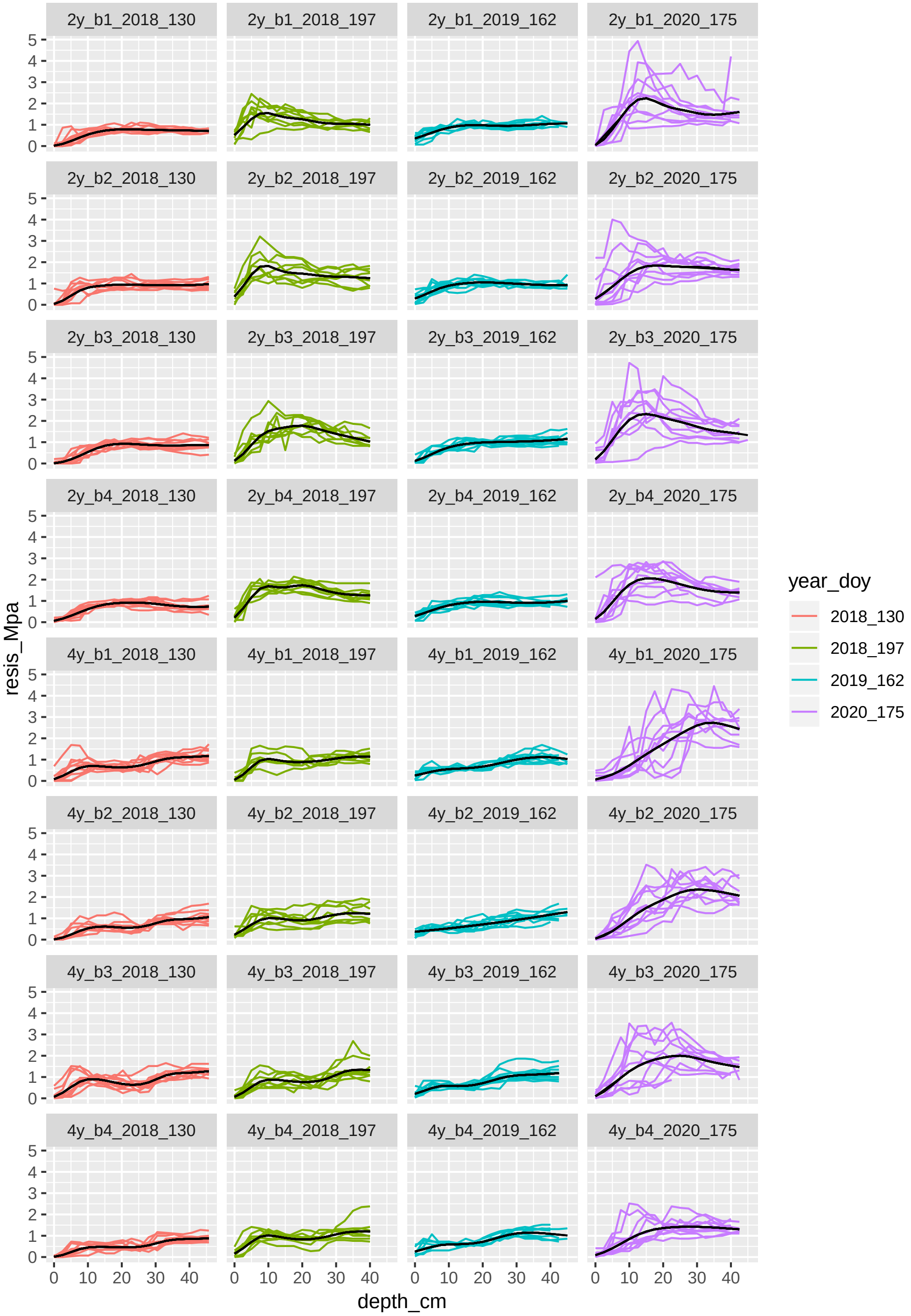
View residuals by group

```r
myd$resid <- mod$residuals

ggplot(data = myd) +
  geom_line(aes(x = depth_cm, y = resid, group = rep_id, color = year_doy)) +
  facet_wrap( ~ trt_block_yr, ncol = length(unique(myd$year_doy)))
```



View data with fitted curves by group

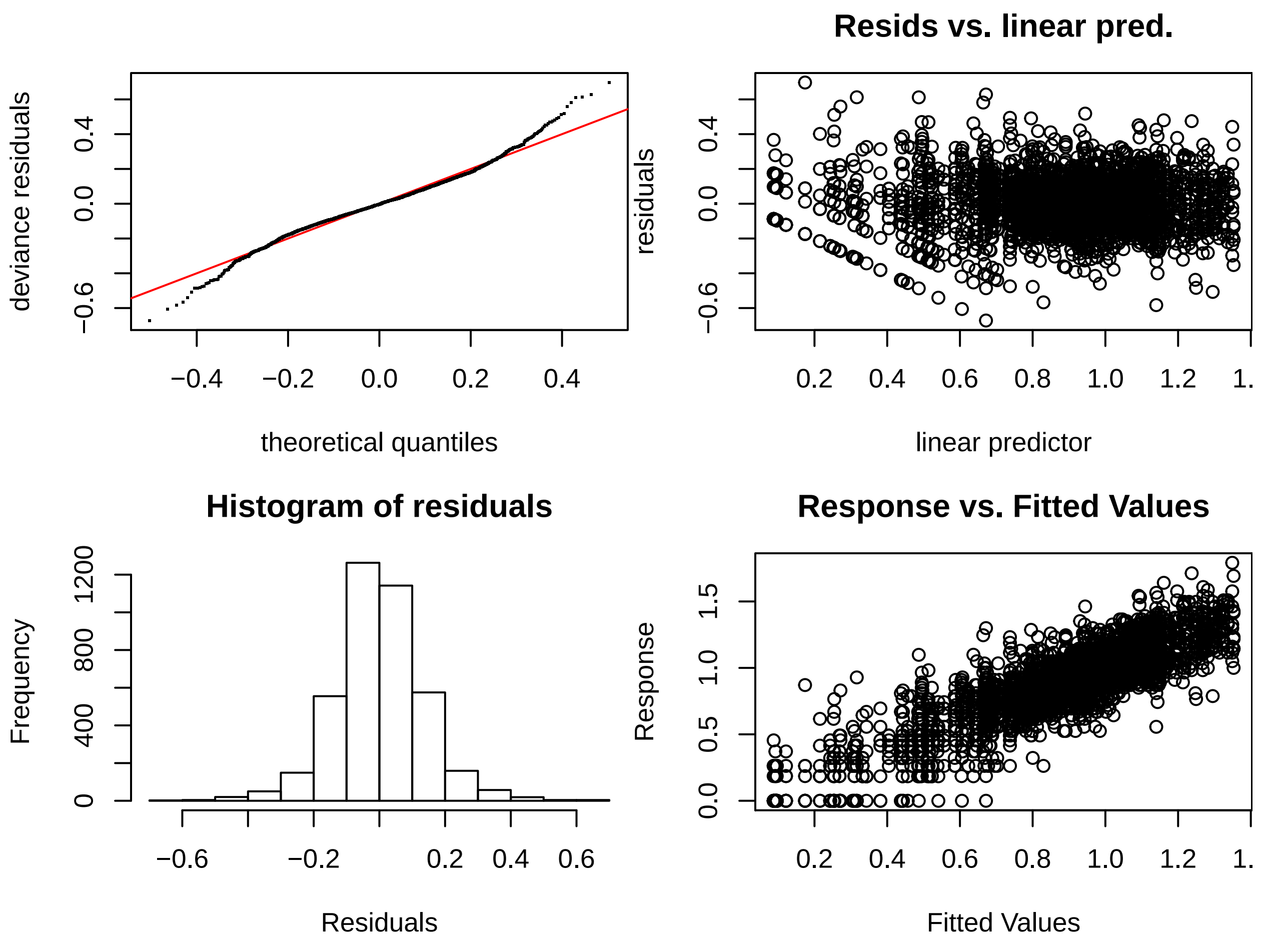
```r
ggplot(data = myd) +
  geom_line(aes(x = depth_cm, y = resis_Mpa, group = rep_id, color = year_doy)) +
  geom_line(aes(x = depth_cm, y = p^2, group = rep_id), color = "black") + # note p^2
  facet_wrap( ~ trt_block_yr, ncol = length(unique(myd$year_doy)))
```



Remove year that is most unlike the others and (re)refit

```r
myd <- filter(myd, year != 2020) # modifying in place because I'm lazy :(
myd$trt_block_yr <- as.factor(as.character(myd$trt_block_yr))

mod <- gam(sqrt(resis_Mpa) ~ s(depth_cm, by = trt_block_yr, bs = "cr", k = 8) + trt_block_yr,
           data = myd, method = "REML")
```

```r
# plot(mod, residuals = TRUE, shade = TRUE)
par(mar = c(4, 4, 3, 0))
mgcv::gam.check(mod)
```



```
##
## Method: REML   Optimizer: outer newton
## full convergence after 5 iterations.
## Gradient range [-0.0005087943,0.0002604082]
## (score -1967.528 & scale 0.01890812).
## Hessian positive definite, eigenvalue range [0.5124608,1977.562].
## Model rank =  192 / 192
##
```

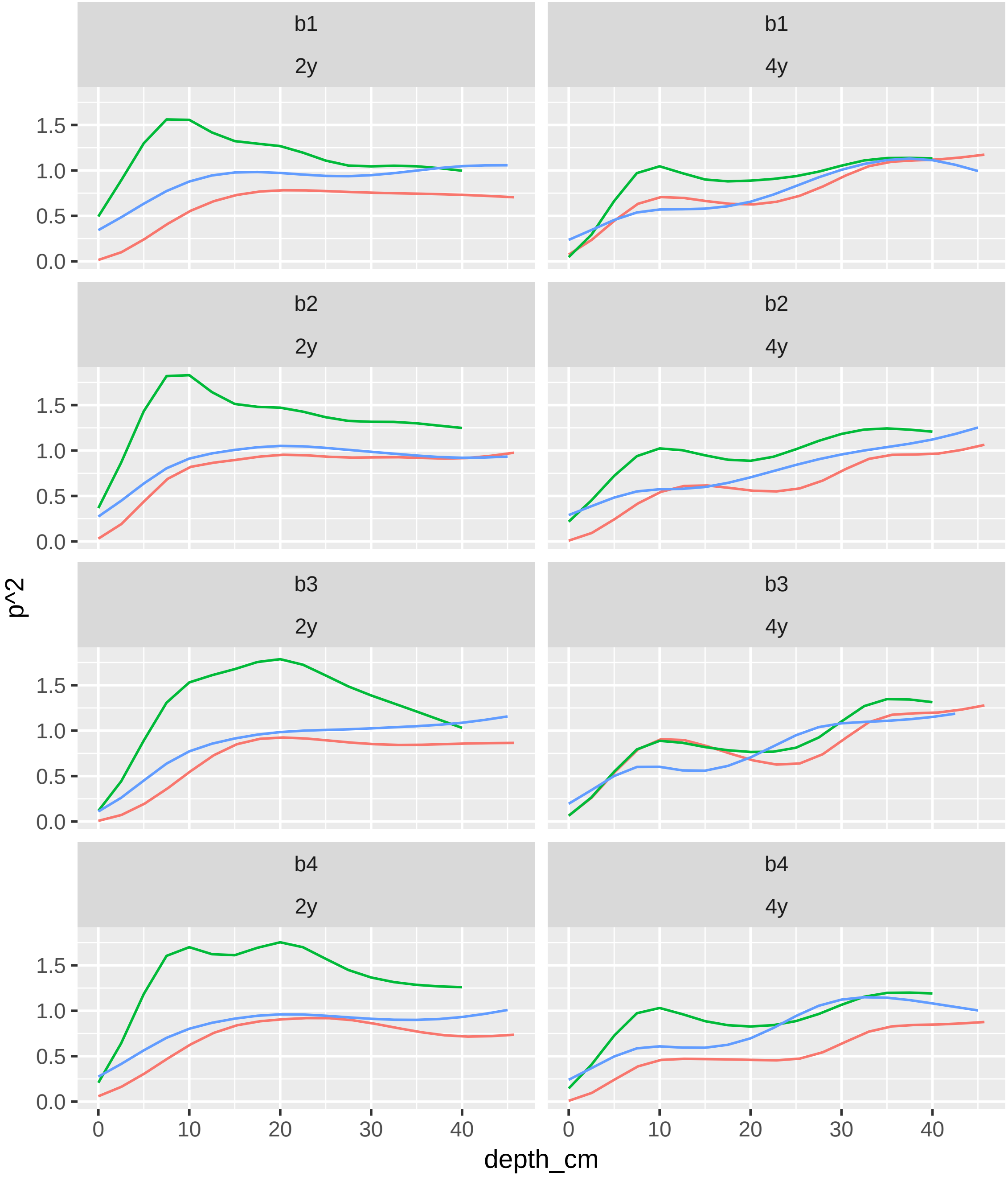
```
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                                         k'  edf k-index p-value
## s(depth_cm):trt_block_yr2y_b1_2018_130 7.00 5.31    1.06       1
## s(depth_cm):trt_block_yr2y_b1_2018_197 7.00 5.63    1.06       1
## s(depth_cm):trt_block_yr2y_b1_2019_162 7.00 4.49    1.06       1
## s(depth_cm):trt_block_yr2y_b2_2018_130 7.00 5.95    1.06       1
## s(depth_cm):trt_block_yr2y_b2_2018_197 7.00 5.89    1.06       1
## s(depth_cm):trt_block_yr2y_b2_2019_162 7.00 4.99    1.06       1
## s(depth_cm):trt_block_yr2y_b3_2018_130 7.00 5.28    1.06       1
## s(depth_cm):trt_block_yr2y_b3_2018_197 7.00 5.58    1.06       1
## s(depth_cm):trt_block_yr2y_b3_2019_162 7.00 5.22    1.06       1
## s(depth_cm):trt_block_yr2y_b4_2018_130 7.00 5.09    1.06       1
## s(depth_cm):trt_block_yr2y_b4_2018_197 7.00 5.88    1.06       1
## s(depth_cm):trt_block_yr2y_b4_2019_162 7.00 4.70    1.06       1
## s(depth_cm):trt_block_yr4y_b1_2018_130 7.00 6.02    1.06       1
## s(depth_cm):trt_block_yr4y_b1_2018_197 7.00 5.79    1.06       1
## s(depth_cm):trt_block_yr4y_b1_2019_162 7.00 5.10    1.06       1
## s(depth_cm):trt_block_yr4y_b2_2018_130 7.00 5.95    1.06       1
## s(depth_cm):trt_block_yr4y_b2_2018_197 7.00 5.30    1.06       1
## s(depth_cm):trt_block_yr4y_b2_2019_162 7.00 4.68    1.06       1
## s(depth_cm):trt_block_yr4y_b3_2018_130 7.00 6.33    1.06       1
## s(depth_cm):trt_block_yr4y_b3_2018_197 7.00 5.62    1.06       1
## s(depth_cm):trt_block_yr4y_b3_2019_162 7.00 5.62    1.06       1
## s(depth_cm):trt_block_yr4y_b4_2018_130 7.00 6.00    1.06       1
## s(depth_cm):trt_block_yr4y_b4_2018_197 7.00 5.58    1.06       1
## s(depth_cm):trt_block_yr4y_b4_2019_162 7.00 5.32    1.06       1
```

I’m happy with the `edf`, `k-index`, and p-values here, so we can visualize the model and residuals, and then use the model for inference.

(If you don’t want to throw away the 2020 data entirely, you could fit another GAM to just those data.)

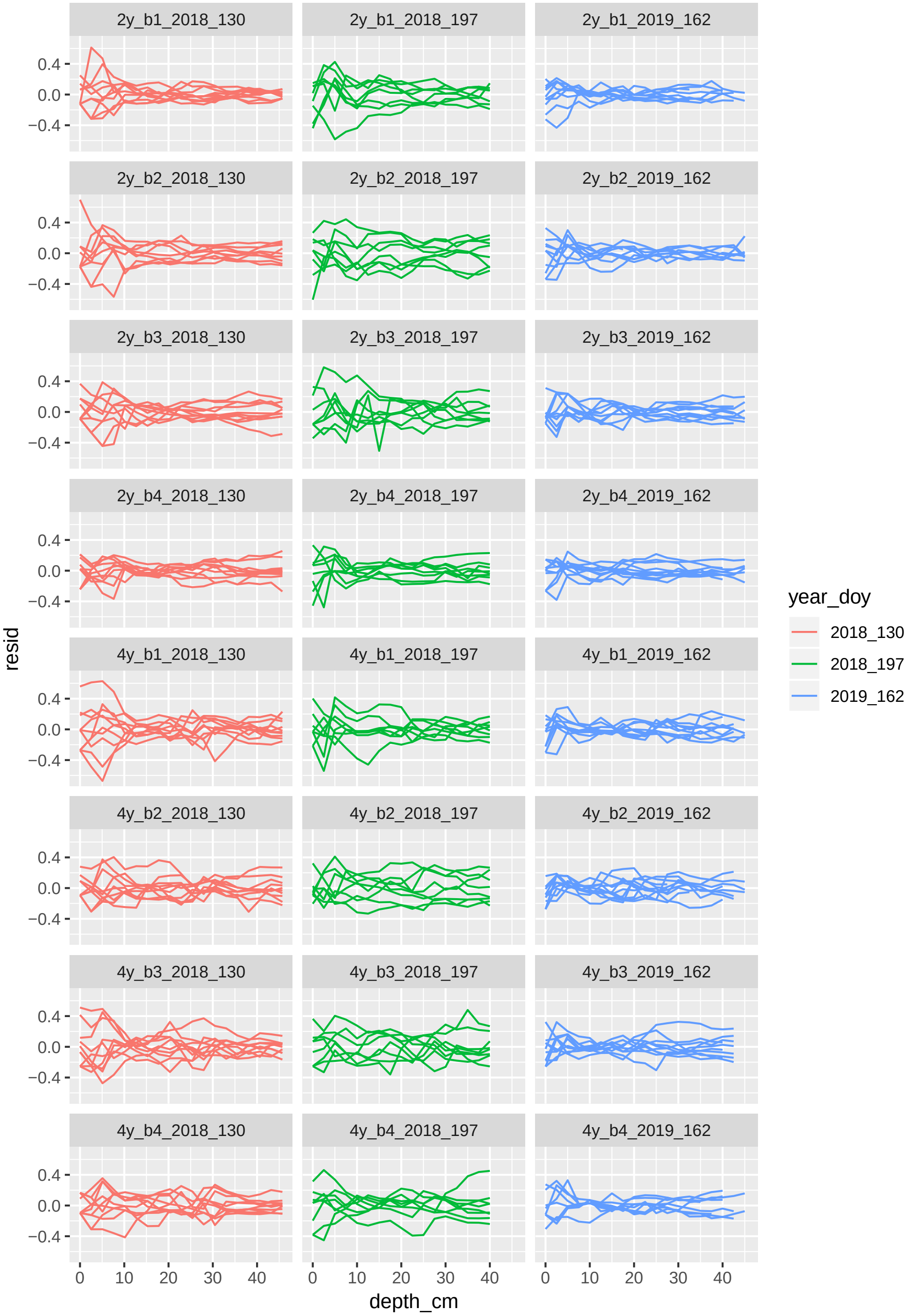
View fitted model by group

```
# view the 24 fitted curves to visually inspect differences
myd$p <- predict(mod)

ggplot(data = myd) +
  geom_line(aes(x = depth_cm, y = p^2, color = year_doy, group = year_doy)) + # note p^2
  facet_wrap( ~ block + rot_trt, ncol = 2) +
  guides(color = FALSE)
```



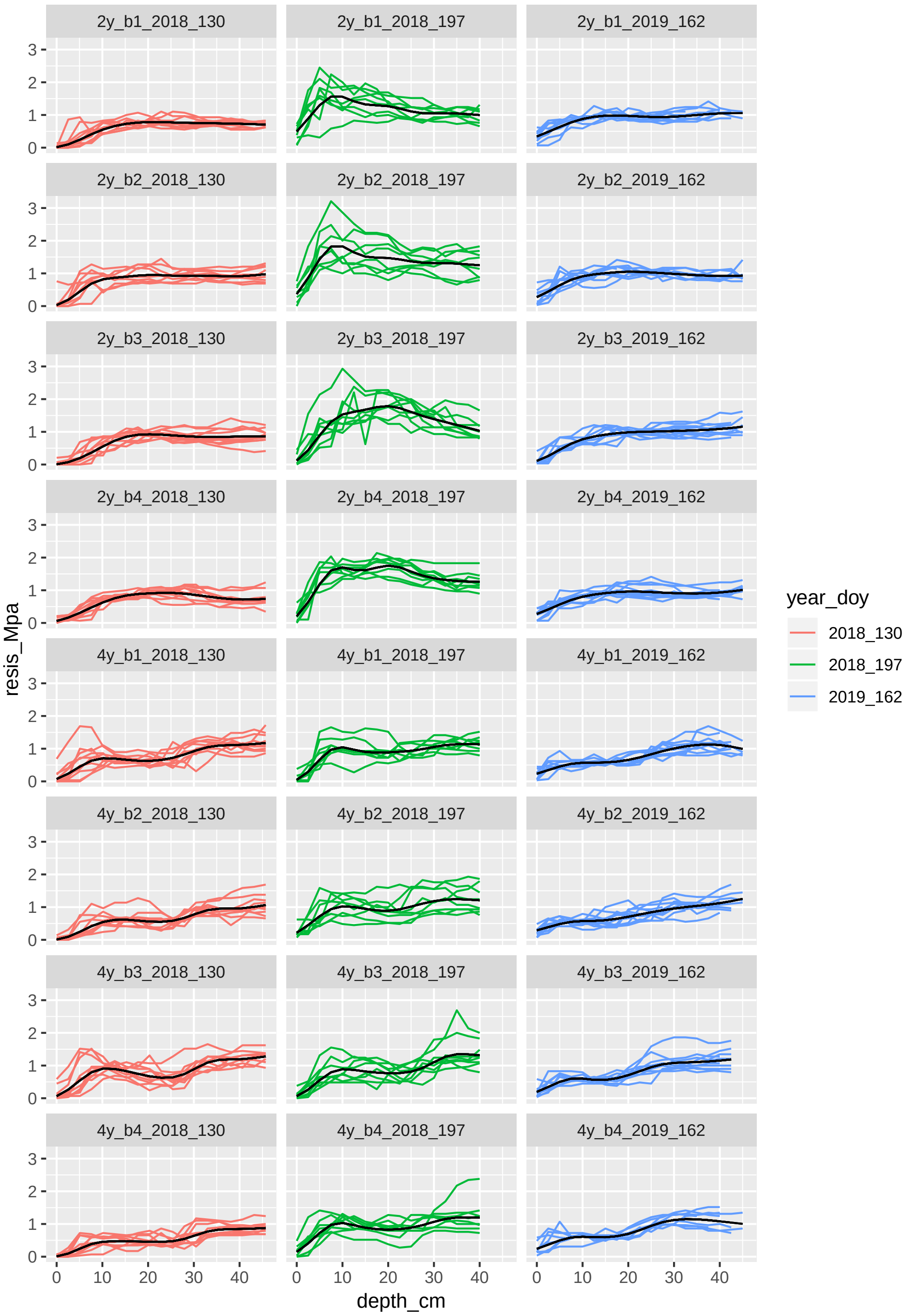
View residuals by group

```r
myd$resid <- mod$residuals

ggplot(data = myd) +
  geom_line(aes(x = depth_cm, y = resid, group = rep_id, color = year_doy)) +
  facet_wrap( ~ trt_block_yr, ncol = length(unique(myd$year_doy)))
```



View data with fitted curves by group

```r
ggplot(data = myd) +
  geom_line(aes(x = depth_cm, y = resis_Mpa, group = rep_id, color = year_doy)) +
  geom_line(aes(x = depth_cm, y = p^2, group = rep_id), color = "black") + # note p^2
  facet_wrap( ~ trt_block_yr, ncol = length(unique(myd$year_doy)))
```



Differences between treatments

To test for differences between treatments, we'll use the following function from https://fromthebottomoftheheap.net/2017/10/10/difference-splines-i/:

```r
smooth_diff <- function(model, newdata, f1, f2, var, alpha = 0.05,
                        unconditional = FALSE) {
    xp <- predict(model, newdata = newdata, type = 'lpmatrix')
    c1 <- grepl(f1, colnames(xp))
    c2 <- grepl(f2, colnames(xp))
    r1 <- newdata[[var]] == f1
    r2 <- newdata[[var]] == f2
    ## difference rows of xp for data from comparison
    X <- xp[r1, ] - xp[r2, ]
    ## zero out cols of X related to splines for other lochs
    X[, ! (c1 | c2)] <- 0
    ## zero out the parametric cols
    X[, !grepl('^s\\(', colnames(xp))] <- 0
    dif <- X %*% coef(model)
    se <- sqrt(rowSums((X %*% vcov(model, unconditional = unconditional)) * X))
    crit <- qt(alpha/2, df.residual(model), lower.tail = FALSE)
    upr <- dif + (crit * se)
    lwr <- dif - (crit * se)
    data.frame(pair = paste(f1, f2, sep = '-'),
               diff = dif,
               se = se,
               upper = upr,
               lower = lwr)
}
```

We want to compare the differences between treatments, within blocks and years. First, we set up a data frame that pairs the variable levels between treatments, within blocks and year/doy values.

```r
# get all combinations of block and year/doy
base_str <- as.vector(sapply(unique(myd$block), function(s) {
  paste(s, unique(myd$year_doy), sep = "_")
  }))

# add trt to the front of each base str, yielding pairwise df
var_df <- data.frame(var1 = paste("2y", base_str, sep = "_"),
                     var2 = paste("4y", base_str, sep = "_"),
                     stringsAsFactors = FALSE)

head(var_df)
```

var1	var2
2y_b1_2018_130	4y_b1_2018_130
2y_b1_2018_197	4y_b1_2018_197
2y_b1_2019_162	4y_b1_2019_162
2y_b2_2018_130	4y_b2_2018_130
2y_b2_2018_197	4y_b2_2018_197
2y_b2_2019_162	4y_b2_2019_162

Next, we create a data frame for prediction that contains the depth values and levels of `trt_block_yr`.

```
newdata <- expand.grid(depth_cm = unique(myd$depth_cm),
                       trt_block_yr = levels(myd$trt_block_yr))

out <- purrr::map_dfr(1:nrow(var_df), function(i) {
  d <- smooth_diff(mod, newdata,
               f1 = var_df[i,1], f2 = var_df[i,2],
               var = "trt_block_yr")
  d$pair <- as.character(d$pair) # prevent map_dfr from combining factors
  return(d)
})

comp <- cbind(depth_cm = unique(myd$depth_cm), out) # add depth values
comp$pair <- as.factor(comp$pair) # make this a factor again for ggplot2
head(comp)
```

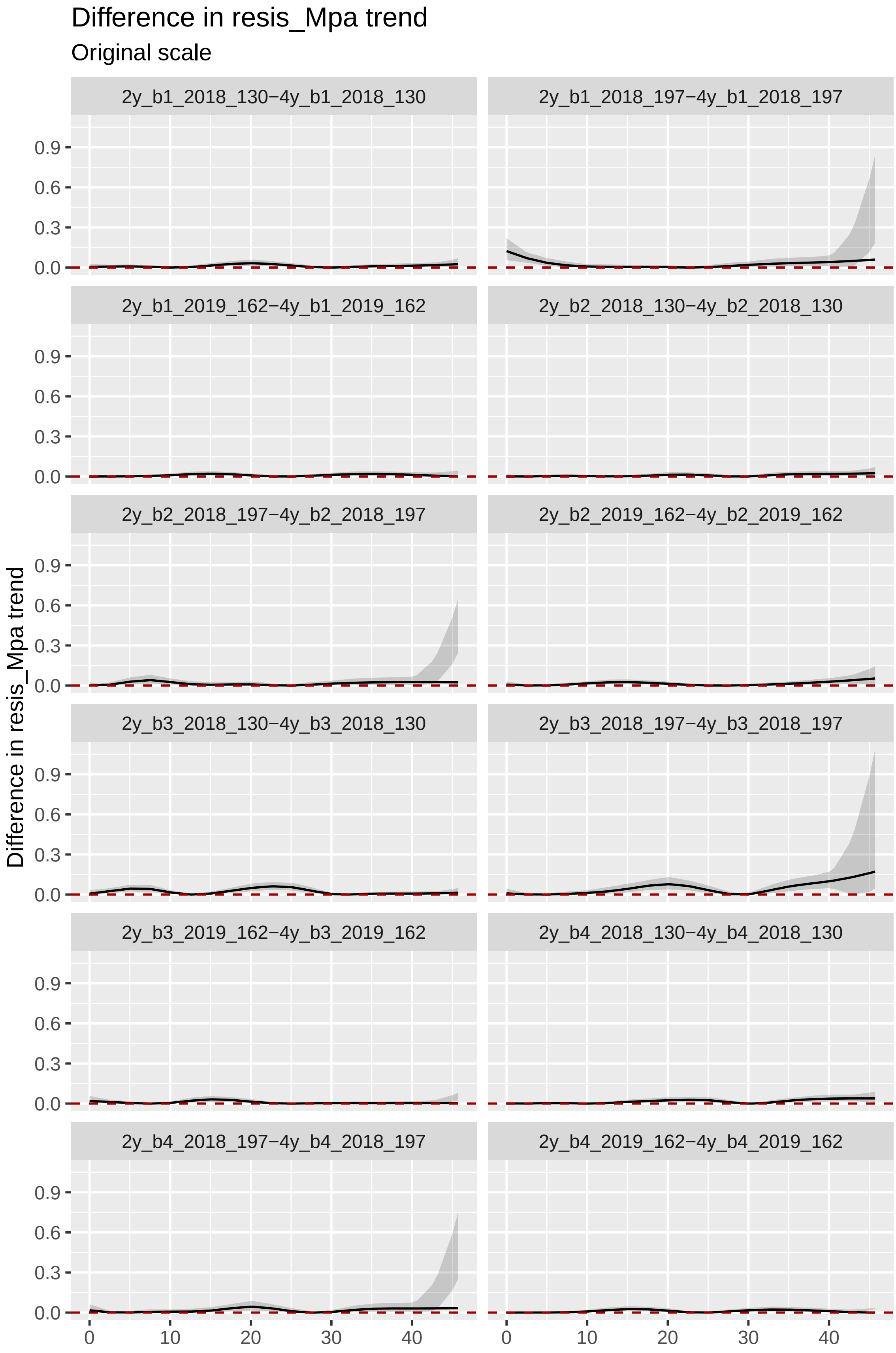
depth_cm	pair	diff	se	upper	lower
0.00	2y_b1_2018_130-4y_b1_2018_130	-0.0646513	0.0496994	0.0327884	-0.1620910
2.54	2y_b1_2018_130-4y_b1_2018_130	-0.0854128	0.0297256	-0.0271334	-0.1436922
5.08	2y_b1_2018_130-4y_b1_2018_130	-0.0915963	0.0303833	-0.0320273	-0.1511653
7.62	2y_b1_2018_130-4y_b1_2018_130	-0.0686242	0.0322925	-0.0053121	-0.1319363
10.16	2y_b1_2018_130-4y_b1_2018_130	-0.0097649	0.0280663	0.0452613	-0.0647911
12.70	2y_b1_2018_130-4y_b1_2018_130	0.0639127	0.0321327	0.1269115	0.0009138

First, remember that the model is predicting sqrt(resis_Mpa), so we should transform the predictions and confidence intervals back to their original scale.

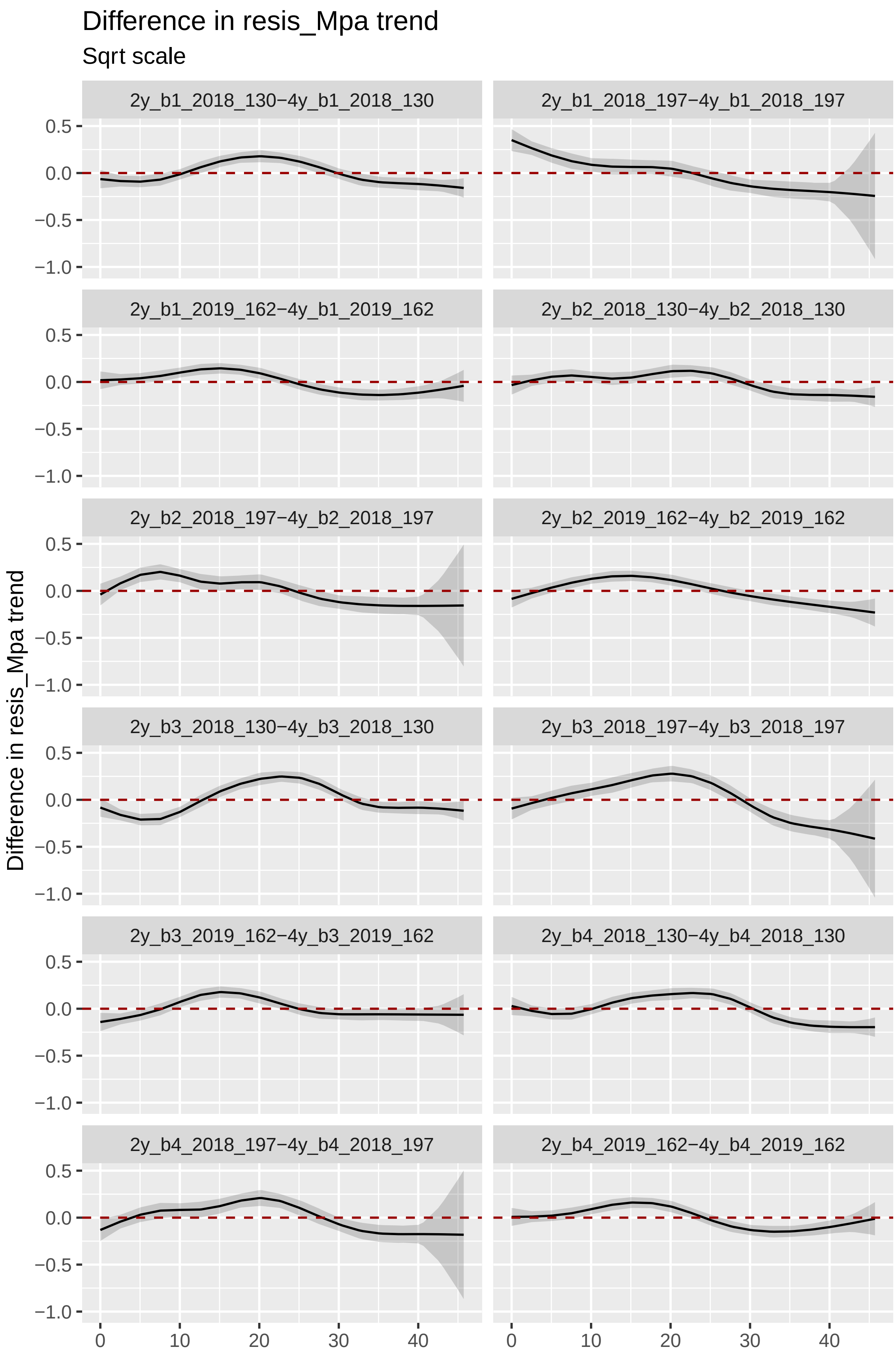
```
comp$diff2 <- comp$diff^2
comp$upper2 <- comp$upper^2
comp$lower2 <- comp$lower^2
```

Now, we can plot the difference between treatments, with associated confidence bands, for each block and year/doy combination. Any depth values where the shading does not cross the dashed red line (at diff = 0) are points where the treatment resistances differ significantly.

```
ggplot(comp, aes(x = depth_cm, y = diff2, group = pair)) +
    geom_ribbon(aes(ymin = lower2, ymax = upper2), alpha = 0.2) +
    geom_line() +
    geom_hline(aes(yintercept = 0), colour="#990000", linetype="dashed") +
    facet_wrap(~ pair, ncol = 2) +
    labs(x = NULL, y = 'Difference in resis_Mpa trend') +
  ggtitle("Difference in resis_Mpa trend", subtitle = "Original scale")
```


Difference in resis_Mpa trend
Original scale
Difference in resis_Mpa trend
2y_b1_2018_130−4y_b1_2018_130
2y_b1_2018_197−4y_b1_2018_197
2y_b1_2019_162−4y_b1_2019_162
2y_b2_2018_130−4y_b2_2018_130
2y_b2_2018_197−4y_b2_2018_197
2y_b2_2019_162−4y_b2_2019_162
2y_b3_2018_130−4y_b3_2018_130
2y_b3_2018_197−4y_b3_2018_197
2y_b3_2019_162−4y_b3_2019_162
2y_b4_2018_130−4y_b4_2018_130
2y_b4_2018_197−4y_b4_2018_197
2y_b4_2019_162−4y_b4_2019_162
0.9
0.6
0.3
0.0
0
10
20
30
40

```
ggplot(comp, aes(x = depth_cm, y = diff, group = pair)) +
    geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.2) +
    geom_line() +
    geom_hline(aes(yintercept = 0), colour="#990000", linetype="dashed") +
    facet_wrap(~ pair, ncol = 2) +
    labs(x = NULL, y = 'Difference in resis_Mpa trend') +
  ggtitle("Difference in resis_Mpa trend", subtitle = "Sqrt scale")
```


Difference in resis_Mpa trend
Sqrt scale
2y_b1_2018_130−4y_b1_2018_130
2y_b1_2018_197−4y_b1_2018_197
2y_b1_2019_162−4y_b1_2019_162
2y_b2_2018_130−4y_b2_2018_130
2y_b2_2018_197−4y_b2_2018_197
2y_b2_2019_162−4y_b2_2019_162
2y_b3_2018_130−4y_b3_2018_130
2y_b3_2018_197−4y_b3_2018_197
2y_b3_2019_162−4y_b3_2019_162
2y_b4_2018_130−4y_b4_2018_130
2y_b4_2018_197−4y_b4_2018_197
2y_b4_2019_162−4y_b4_2019_162
Difference in resis_Mpa trend
0.5
0.0
−0.5
−1.0
0
10
20
30
40