Analysis

An analysis using LiDAR data to detect moose browsing effects, with ground truthing

```
library(readr)
library(ggplot2)
library(glmmTMB)
```

Get compiled dataset (see compile.R)

```
dat <- read_csv("../data/compiledDataset.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   locality_and_treatment = col_character(),
##   LocalityCode = col_character(),
##   LocalityName = col_character(),
##   Treatment = col_character(),
##   resolution_m = col_character(),
##   region = col_character()
## )

## See spec(...) for full column specifications.
```

```
head(dat)
```

```
## # A tibble: 6 x 27
##   locality_and_tr~ LocalityCode LocalityName Treatment Longitude Latitude
##   <chr>            <chr>        <chr>        <chr>         <dbl>    <dbl>
## 1 bratsberg_b      BRB          Bratsberg    B              10.5     63.4
## 2 bratsberg_ub     BRUB         Bratsberg    UB             10.5     63.4
## 3 didrik_holmsen_b DHB          Didrik Holm~ B              11.4     59.9
## 4 didrik_holmsen_~ DHUB         Didrik Holm~ UB             11.4     59.9
## 5 drangedal1_b     1DRB         Drangedal1   B               9.15    59.1
## 6 drangedal1_ub    1DRUB        Drangedal1   UB              9.15    59.1
## # ... with 21 more variables: Clear.cut <dbl>, Year.initiated <dbl>,
## #   LiDAR.data.from.year <dbl>, plot_density_m2 <dbl>, resolution_m <chr>,
## #   region <chr>, Moose2015 <dbl>, Reddeer2015 <dbl>, Roedeer2015 <dbl>,
## #   YrsSinceExclosure <dbl>, field_mean <dbl>, field_median <dbl>, mn <dbl>,
## #   md <dbl>, sd <dbl>, min <dbl>, max <dbl>, first_qu.25. <dbl>,
## #   third_qu.75. <dbl>, mad <dbl>, prod <dbl>
```

A quick data check

```
table(dat$Treatment, dat$Clear.cut)
```

```
##
##      2000 2002 2003 2004 2005 2006 2007 2008 2009
##   B     1    4    3    7    8    4   10    7    1
##   UB    1    4    3    7    8    4   10    7    1
```

```
table(dat$Year.initiated, dat$LiDAR.data.from.year)
```

```
##
##        2010 2011 2013 2015 2016 2017 2018 2019
##   2007    0    0    0    0    0    0    0    6
##   2008    4    4    0   18    2    2    0    0
##   2009    0    0    0    0    4   24    0    0
##   2010    0    0    0    0    4    4    4    0
##   2011    0    0    2    0    4    0    0    8
```

```
table(dat$plot_density_m2, dat$resolution_m)
```

```
##
##     0,25 0,5
##   2    2  54
##   5   32   2
```

Something odd there…

```
table(dat$region, dat$Treatment)
```

```
##
##               B UB
##   Hedmark   16 16
##   Telemark  14 14
##   Trondelag 15 15
```

```
table(dat$LocalityName, dat$Treatment)
```

```
##
##                          B UB
##   Bratsberg              1  1
##   Didrik Holmsen         1  1
##   Drangedal1             1  1
##   Drangedal3             1  1
##   Drangedal4             1  1
##   Eidskog                1  1
##   Fet 3                  1  1
##   Fritsoe1               1  1
##   Fritsoe2               1  1
##   Furesdal              1  1
##   Halvard Pramhus       1  1
##   Hi_tydal              1  1
##   Kongsvinger 1         1  1
##   Kongsvinger 2         1  1
##   Kviteseid1            1  1
##   Kviteseid2            1  1
##   Kviteseid3            1  1
##   Maarud 1              1  1
##   Maarud 2              1  1
##   Maarud 3              1  1
##   Malvik                1  1
##   namdalseid_1kub       1  1
##   Nes 1                 1  1
##   Nes 2                 1  1
##   Nome_Cappelen1        1  1
##   Nome_Cappelen2        1  1
##   Notodden3             1  1
##   Notodden5             1  1
##   Notodden6             1  1
##   Nsb_Verdal            1  1
##   Selbu_Flub            1  1
##   Selbu_kl              1  1
##   Selbu_Sl              1  1
##   Singsaas              1  1
##   Sl_Tydal              1  1
##   Soerum                1  1
##   Stangeskovene Aurskog 1  1
##   Stangeskovene Eidskog 1  1
##   steinkjer_1BBb        1  1
##   steinkjer_2BBb        1  1
##   Stig Dahlen           1  1
##   Sub_Namdalseid        1  1
##   Truls Holm            1  1
##   verdal_1vb            1  1
##   verdal_2VB            1  1
```

Looks good.

Canopy growth per year

Lets first compute canopy growth per year since exclosure

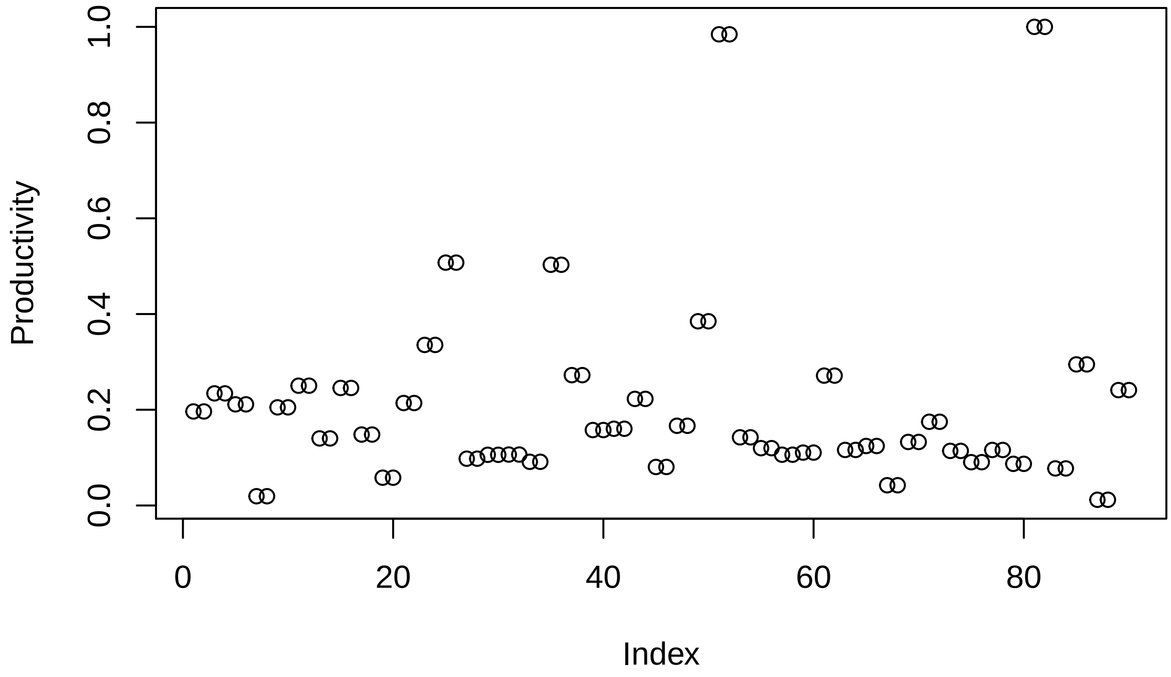
```
dat$canopygrowth <- dat$md/dat$YrsSinceExclosure
summary(dat$canopygrowth)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
## 0.001214 0.022698 0.045375 0.096275 0.165922 0.420857
```

The numbers are in meters I'm pretty sure

One of the first things to decide on is what to do with productivity, as there are two outliers:

```
plot(dat$prod, ylab="Productivity")
```

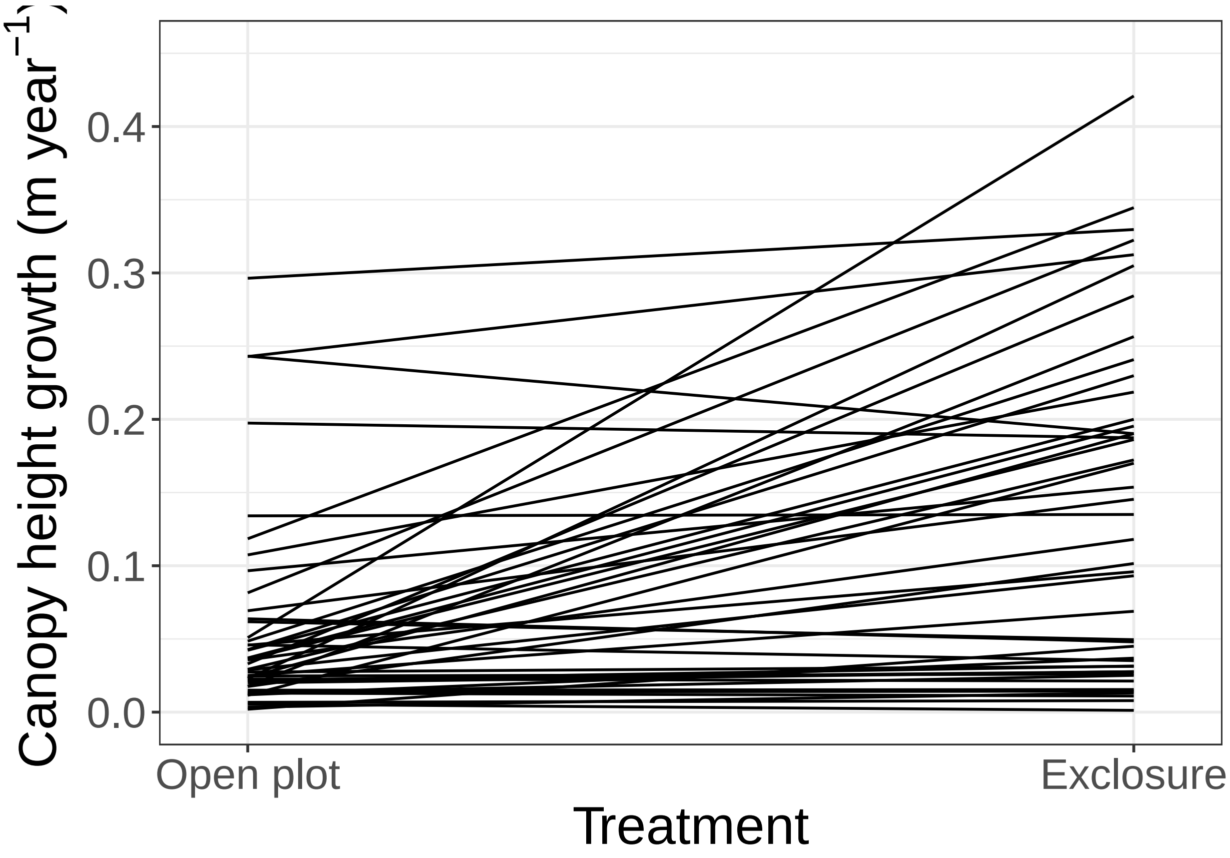


```
dat[dat$prod>0.6,c("LocalityName", "region", "prod")]
```

```
## # A tibble: 4 x 3
##   LocalityName region   prod
##   <chr>        <chr>   <dbl>
## 1 Nes 2        Hedmark 0.984
## 2 Nes 2        Hedmark 0.984
## 3 Stig Dahlen  Hedmark 1
## 4 Stig Dahlen  Hedmark 1
```

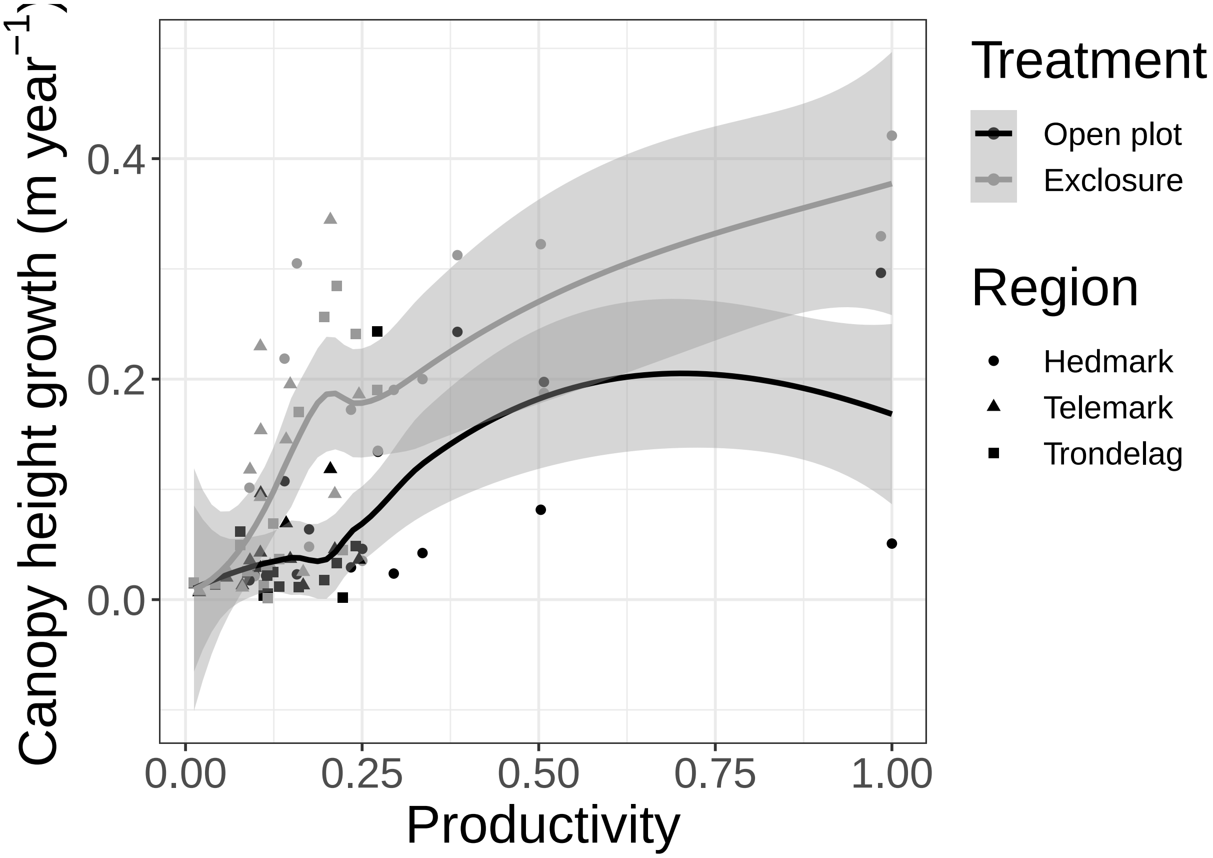
These are two sites in Hedmark that probably really are very productive, although there could have been serious sampling error due to chance. But they are legitimate, and we should be careful not to drop them too willingly. We can use the log of the productivity is residuals are acting strange.

```
dat$Treatment <- as.factor(dat$Treatment)
levels(dat$Treatment) <- c('Open plot','Exclosure')
ggplot(dat, aes(x=Treatment, y=canopygrowth, group=LocalityName))+
  geom_line()+
  labs(y=expression(paste('Canopy height growth (m year'^'-1', ')')), x='Treatment')+
  scale_linetype_manual(breaks = c("Exclosure", "Open plot"),
                        labels = c("Open plots", "Exclosures"), values=c(1,2))+
  scale_x_discrete(limits = c('Open plot', 'Exclosure'),
                   breaks = c(’Open plot’, ’Exclosure’), expand = c(0.1,0))+
  theme_bw()+
  theme(text = element_text(size = 20))+
  ylim(0, 0.45)
```



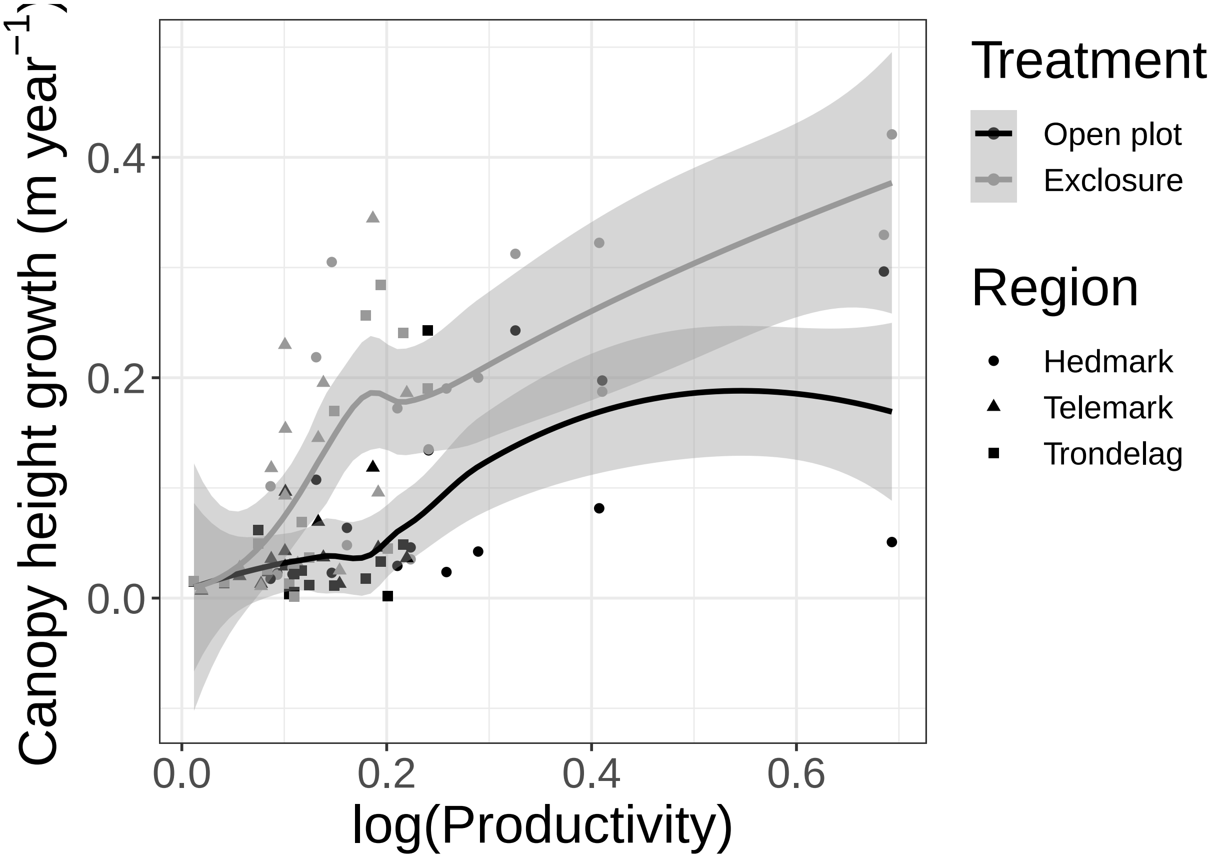
Lets use loess as it makes less assumptions about the shape of the relationship:

```
ggplot(data = dat,
                    aes(x = prod, y = canopygrowth))+
  geom_point(aes(colour= Treatment, shape=region))+
   geom_smooth(aes(colour= Treatment),
               method = "loess", formula =  ’y ~ x’)+
  labs(y=expression(paste(’Canopy height growth (m year’^’-1’, ’)’)), x=’Productivity’)+
  theme_bw()+
  scale_color_manual(values = c("gray0", "gray60"))+
  labs(colour="Treatment", shape="Region")+
  theme(text = element_text(size = 20))+
  #ylim(0, 0.4)+
  theme(legend.position = ’right’,
                             legend.justification = c("left", "top"),
                             legend.box.just = "left",
                             #legend.margin = margin(5, 5, 5, 5),
                             legend.text = element_text(size=12))
```



The highly productive sites are affecting this picture quite a bit I’d imagine. Let’s try with a log-transformation:

```
dat$prod_l <- log(dat$prod+1)
ggplot(data = dat,
                    aes(x = prod_l, y = canopygrowth))+
  geom_point(aes(colour= Treatment, shape=region))+
   geom_smooth(aes(colour= Treatment),
               method = "loess", formula =  ’y ~ x’)+
  labs(y=expression(paste(’Canopy height growth (m year’^’-1’, ’)’)), x=’log(Productivity)’)+
  theme_bw()+
  scale_color_manual(values = c("gray0", "gray60"))+
  labs(colour="Treatment", shape="Region")+
  theme(text = element_text(size = 20))+
  #ylim(0, 0.4)+
  theme(legend.position = ’right’,
                             legend.justification = c("left", "top"),
                             legend.box.just = "left",
                             #legend.margin = margin(5, 5, 5, 5),
                             legend.text = element_text(size=12))
```

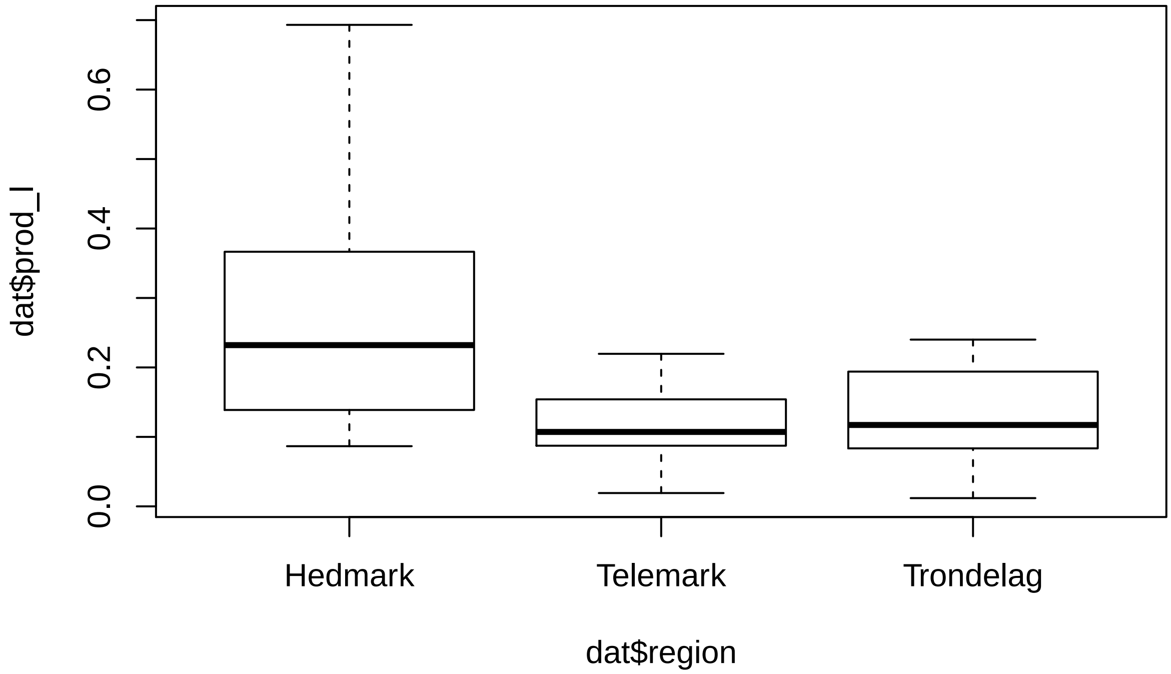


That's perhaps a little better. There's a relationship there it seems like, but perhaps not linear. Also note that all the really productive sites are in Hedmark:

```
table(round(dat$prod, 1), dat$region)
```

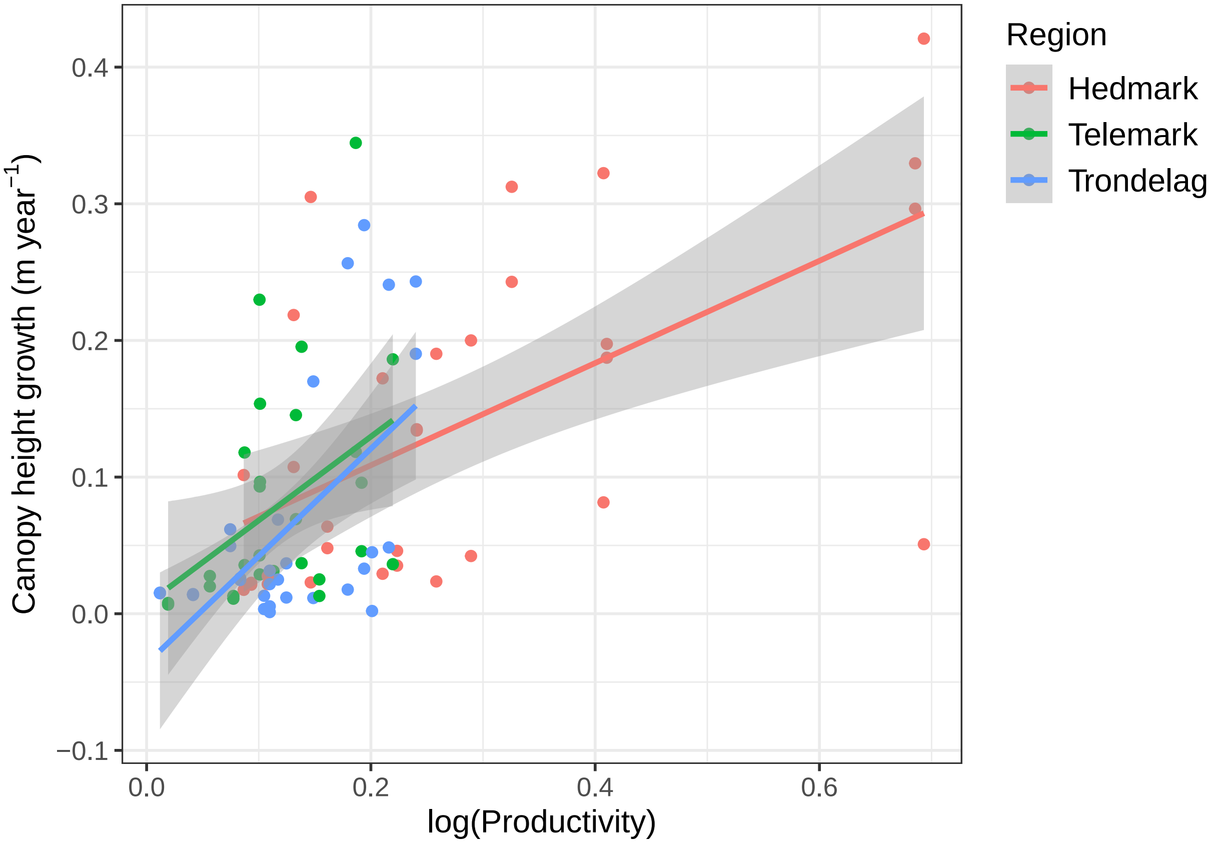
```
##
##       Hedmark Telemark Trondelag
##   0         0        2         4
##   0.1       8       18        14
##   0.2       6        8        10
##   0.3       8        0         2
##   0.4       2        0         0
##   0.5       4        0         0
##   1         4        0         0
```

```
boxplot(dat$prod_l~dat$region)
```



Lets investigate the effect of region a bit more.

```
ggplot(data = dat,
  aes(x = prod_l, y = canopygrowth, colour = region))+
  geom_point()+
  geom_smooth(method = "lm")+
  labs(y=expression(paste('Canopy height growth (m year'^'-1', ')')), x='log(Productivity)')+
  theme_bw()+
  #scale_color_manual(values = c("gray0", "gray50", "grey100"))+
  labs(colour="Region")+
  theme(text = element_text(size = 12))+
  #ylim(0, 0.4)+
  theme(legend.position = 'right',
                             legend.justification = c("left", "top"),
                             legend.box.just = "left",
                             #legend.margin = margin(5, 5, 5, 5),
                             legend.text = element_text(size=12))
```



This plot tells us there is no justification for a random slope model. Also, the two productivity outliers appear less like outliers when only looking at the Hedmark data points.

Summary stats (OBS, don't report these - keep reading)

Let's get some summary stats.

```
m <- tapply(dat$canopygrowth, dat$Treatment, FUN = mean)
# And standard error:
source('se.R')
s <- tapply(dat$canopygrowth, dat$Treatment, FUN = se)
(sumStats <- data.frame("mean canopy growth per year" = round(m, 3),
                        "standard error of the mean" = round(s, 3)))
```

```
##             mean.canopy.growth.per.year standard.error.of.the.mean
## Open plot                         0.056                      0.010
## Exclosure                         0.136                      0.017
```

We need a model to describe the interaction between treatment and productivity. The response is continous and both positive and negative values can occur, so a gaussian familiy would be appropriate.

```
library(glmmTMB) # using this package even though I dont need all its finesse...
dat$region <- as.factor(dat$region)
mod1 <- glmmTMB(canopygrowth ~ Treatment * prod_l + (1|region),
  data = dat,  family = gaussian)
```

```
mod2 <- glmmTMB(canopygrowth ~ Treatment * prod_l ,
  data = dat,  family = gaussian)
```

```
AIC(mod1, mod2)
```

```
##      df       AIC
## mod1  6 -215.4912
## mod2  5 -217.4912
```

Model 2 is slightly better, suggesting we drop the random intercept. What does ANOVA say about this?

```
anova(mod1, mod2)
```

```
## Data: dat
## Models:
## mod2: canopygrowth ~ Treatment * prod_l, zi=~0, disp=~1
## mod1: canopygrowth ~ Treatment * prod_l + (1 | region), zi=~0, disp=~1
##      Df     AIC     BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## mod2  5 -217.49 -204.99 113.75  -227.49
## mod1  6 -215.49 -200.49 113.75  -227.49     0      1          1
```

The models explain exactly the same, so we can drop the random intercept. The only reason for keeping it is if we want to argue it's part of the design, which it is. However, I don't think that's more important than having a parsimoinous model.

```
summary(mod2)
```

```
##  Family: gaussian  ( identity )
## Formula:          canopygrowth ~ Treatment * prod_l
## Data: dat
##
##      AIC      BIC   logLik deviance df.resid
##   -217.5   -205.0    113.7   -227.5       85
##
##
## Dispersion estimate for gaussian family (sigma^2): 0.00467
##
## Conditional model:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)               0.004464   0.016613   0.269  0.78815
## TreatmentExclosure        0.030452   0.023495   1.296  0.19493
## prod_l                    0.287174   0.072826   3.943 8.04e-05 ***
## TreatmentExclosure:prod_l 0.275917   0.102992   2.679  0.00738 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

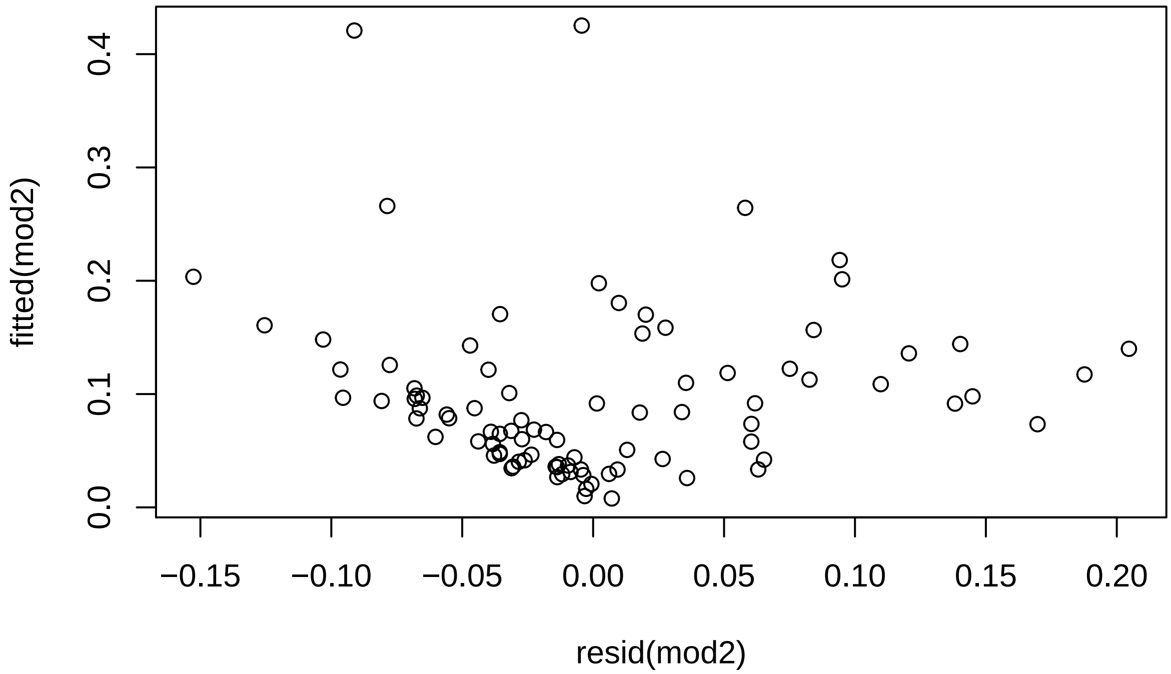
Significant interaction term, so we're keeping this full model. We can also see what the random effect would've give us:

```
ranef(mod1)
```

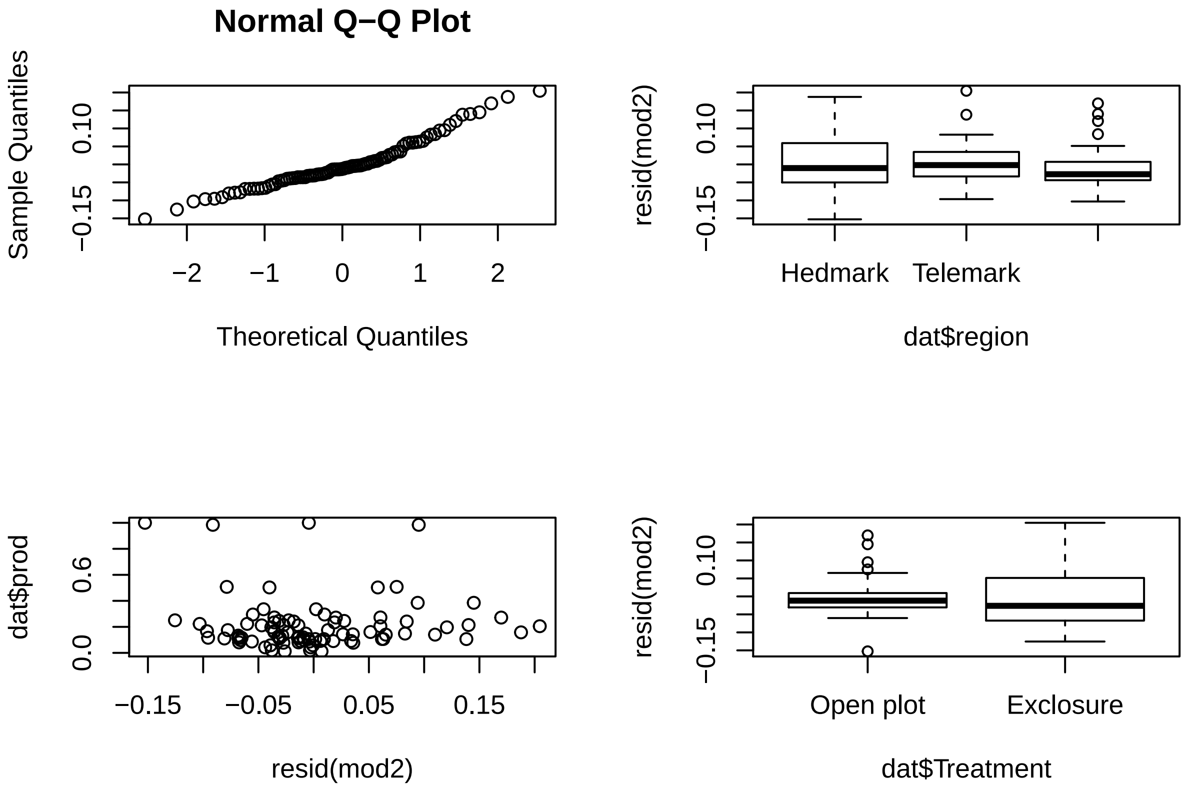
```
## $region
##            (Intercept)
## Hedmark   -1.809701e-12
## Telemark   1.271089e-10
## Trondelag -1.252993e-10
```

, and it's not very much.

```
plot(resid(mod2), fitted(mod2))
```

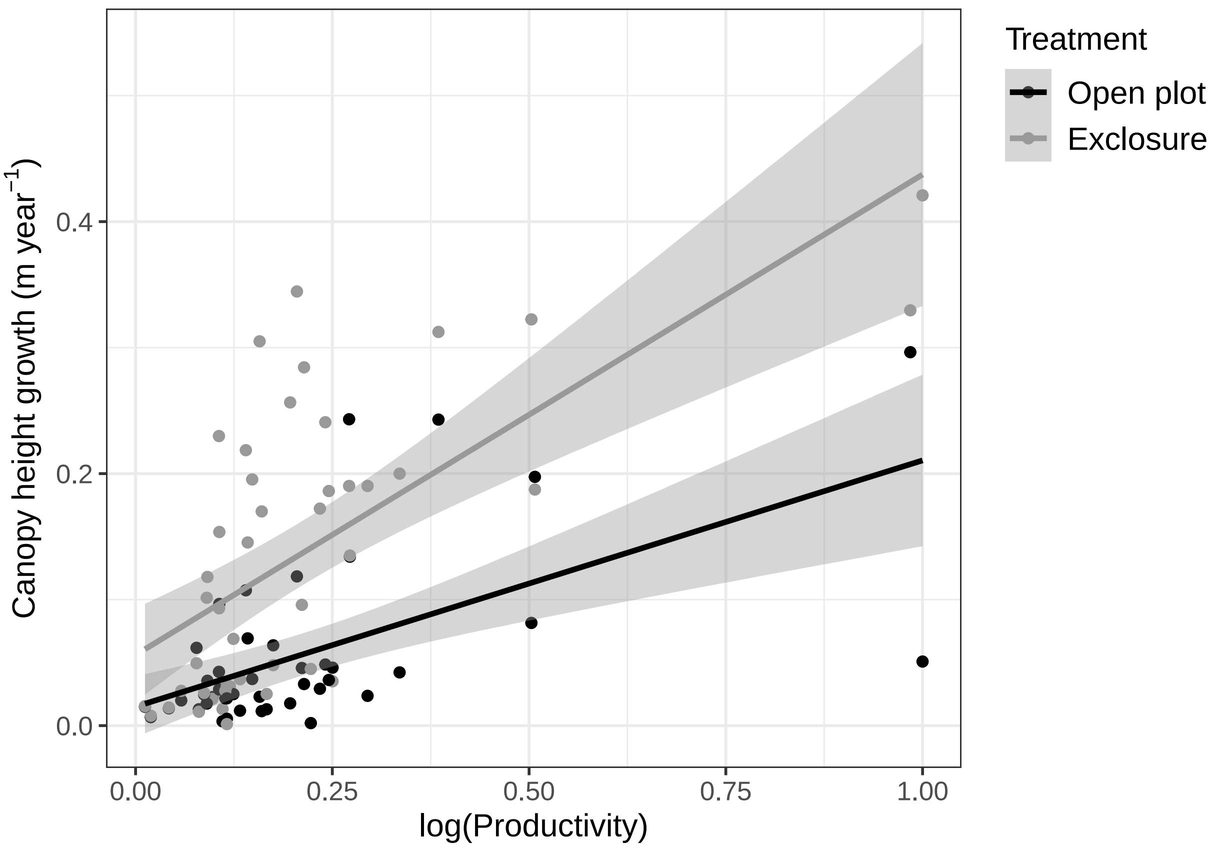


```
par(mfrow=c(2,2))
qqnorm(resid(mod2))
plot(resid(mod2)~ dat$region)
plot(resid(mod2), dat$prod)
plot(resid(mod2)~ dat$Treatment)
```



Looks like everything is in order. The residuals agains productivity at least sho a centering around zero.

```
ggplot(data = dat,
                    aes(x = prod, y = canopygrowth, colour= Treatment))+
  geom_point()+
   geom_smooth(method = "lm")+
  labs(y=expression(paste('Canopy height growth (m year'^'-1', ')')), x='log(Productivity)')+
  theme_bw()+
  scale_color_manual(values = c("gray0", "gray60"))+
  labs(colour="Treatment")+
  theme(text = element_text(size = 12))+
  #ylim(0, 0.4)+
  theme(legend.position = 'right',
                             legend.justification = c("left", "top"),
                             legend.box.just = "left",
                             #legend.margin = margin(5, 5, 5, 5),
                             legend.text = element_text(size=12))
```



So here we're modelling a linear function of productivity. We could try adding a quadratic term.

```
dat$prod_l2 <- dat$prod_l*dat$prod_l
mod3 <- glmmTMB(canopygrowth ~ Treatment * prod_l + prod_l2,
  data = dat,  family = gaussian)
```

```
AIC(mod2, mod3)
```

```
##      df       AIC
## mod2  5 -217.4912
## mod3  6 -219.5023
```

This gave a slight improvement.

```
anova(mod2, mod3)
```

```
## Data: dat
## Models:
## mod2: canopygrowth ~ Treatment * prod_l, zi=~0, disp=~1
## mod3: canopygrowth ~ Treatment * prod_l + prod_l2, zi=~0, disp=~1
##      Df     AIC     BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## mod2  5 -217.49 -204.99 113.75  -227.49
## mod3  6 -219.50 -204.50 115.75  -231.50 4.0111      1     0.0452 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The numbers tell us we should keep it.

Sensitivity analysis

Finally, we should do a sensitivity analysis by removing the outliers and comparing the results that those models give us.

```
dat2 <- dat[dat$prod <0.7,]
dim(dat2)
```

```
## [1] 86 30
```

That deleted the two most productive sites.

```
mod_sens1 <- glmmTMB(canopygrowth ~ Treatment * prod + (1|region),
  data = dat2,  family = gaussian)
mod_sens2 <- glmmTMB(canopygrowth ~ Treatment * prod,
  data = dat2,  family = gaussian)
AIC(mod_sens1, mod_sens2)
```

```
##           df       AIC
## mod_sens1  6 -211.5329
## mod_sens2  5 -213.5329
```

```
mod_sens3 <- glmmTMB(canopygrowth ~ Treatment * prod + I(prod^2),
  data = dat2,  family = gaussian)
AIC(mod_sens2, mod_sens3)
```

```
##           df       AIC
## mod_sens2  5 -213.5329
## mod_sens3  6 -212.8157
```

```
summary(mod_sens2)
```

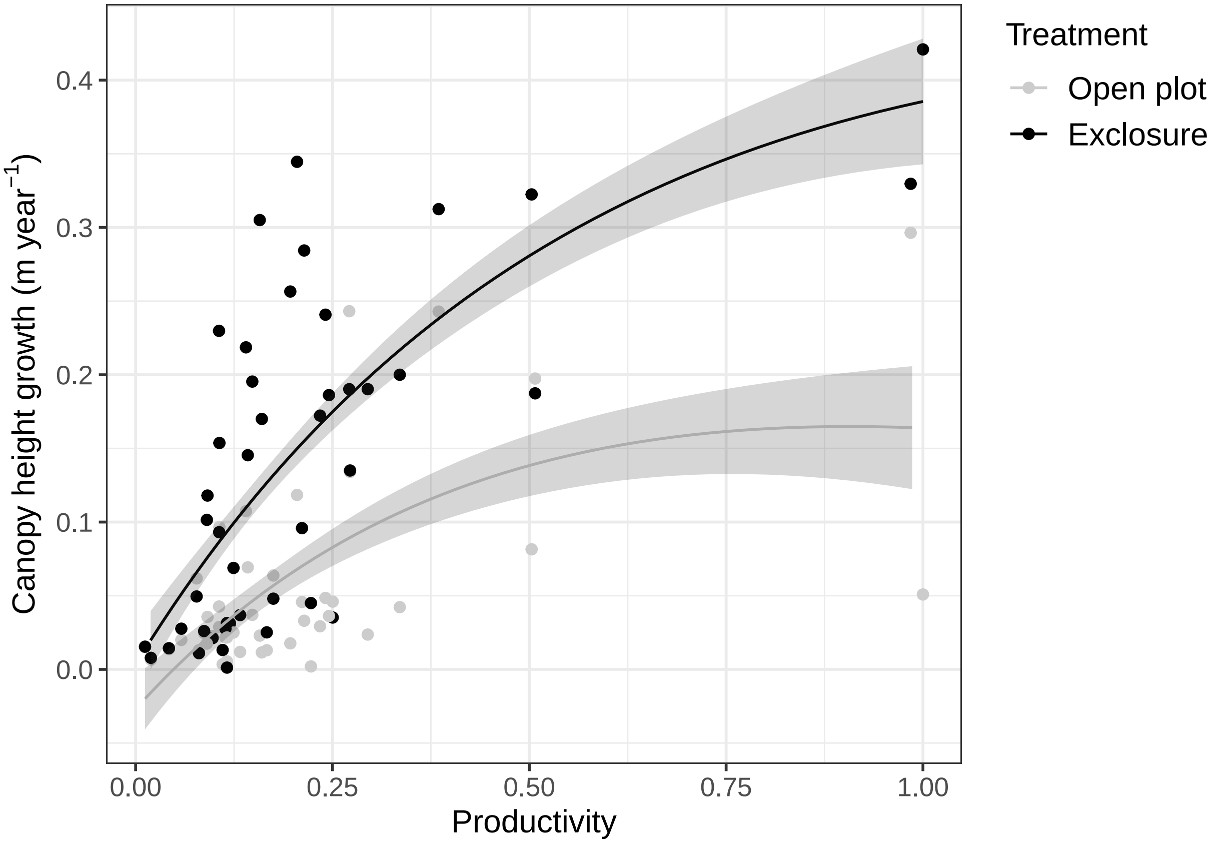
```
##  Family: gaussian  ( identity )
## Formula:          canopygrowth ~ Treatment * prod
## Data: dat2
##
##      AIC      BIC   logLik deviance df.resid
##   -213.5   -201.3    111.8   -223.5       81
##
##
## Dispersion estimate for gaussian family (sigma^2): 0.00435
##
## Conditional model:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -0.003533   0.018914  -0.187 0.851835
## TreatmentExclosure       0.029838   0.026749   1.115 0.264644
## prod                     0.311650   0.091980   3.388 0.000703 ***
## TreatmentExclosure:prod  0.256527   0.130079   1.972 0.048601 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The results are qualitatively the same, which justified keeping the 'outliers' in. (but see discussing below at the end of the MAD chaper)

Plot

```r
tr <- rep(c("Open plot", "Exclosure"), times=50, each=1)
prod_l <- seq(from = min(dat$prod_l), to=max(dat$prod_l), length.out = 100)
prod_l2 <- prod_l^2

pred <- predict(mod3, list(Treatment= tr,
                           prod_l   = prod_l,
                           prod_l2  = prod_l2),
                           se.fit   = TRUE)
pred2 <- data.frame(Treatment = tr,
                    prod_l    = prod_l,
                    pred      = pred$fit,
                    se        = pred$se.fit)
```

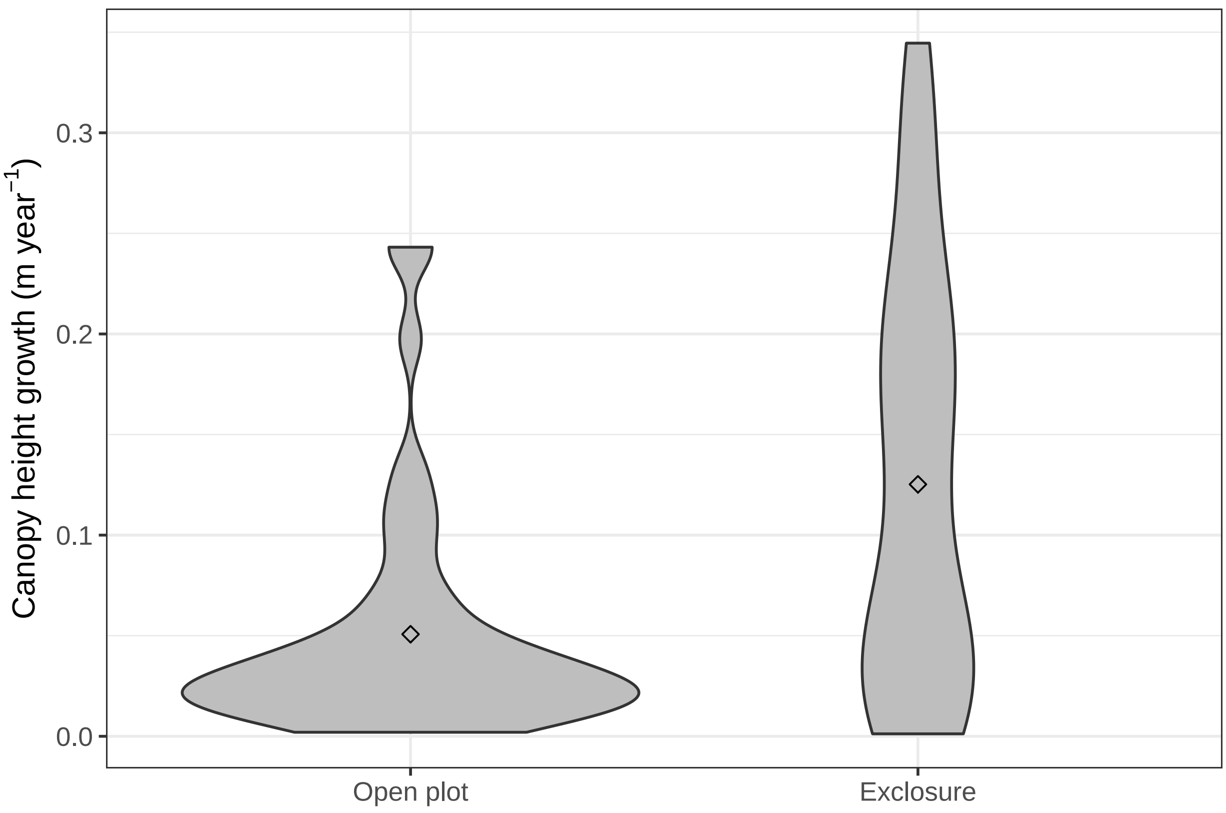
```r
ggplot()+
  geom_point(data = dat,aes(x = exp(prod_l)-1, y = canopygrowth, colour= Treatment))+
  geom_line(data=pred2, aes(x = exp(prod_l)-1, y=pred, colour = Treatment))+
  geom_ribbon(data=pred2, aes(x = exp(prod_l)-1,
                              ymin=pred-se,
                              ymax=pred+se,
                              group = Treatment),
                              alpha=0.2,
                              linetype="blank")+
  labs(y=expression(paste(’Canopy height growth (m year’^’-1’, ’)’)), x=’Productivity’)+
  theme_bw()+
  scale_color_manual(values = c("gray80", "black"))+
  labs(colour="Treatment")+
  theme(text = element_text(size = 12))+
  theme(legend.position = ’right’,
                             legend.justification = c("left", "top"),
                             legend.box.just = "left",
                             #legend.margin = margin(5, 5, 5, 5),
                             legend.text = element_text(size=12))
```



I have backtransformed the productivity axis in this plot. The ribbons are +-SE, but perhaps 95%CIs are better, or +-1.96SE.

We can also make these nice violin plots, even though the main pot should show the interaction effect like the plot above does.

```
ggplot(data = dat2, aes(x = Treatment, y = canopygrowth))+
  geom_violin(fill = "grey")+
  theme_bw()+
  theme(text = element_text(size = 12))+
  labs(y=expression(paste('Canopy height growth (m year'^'-1', ')')), x='')+
  stat_summary(fun.y=mean, geom="point", shape=23, size=2)
```



MAD

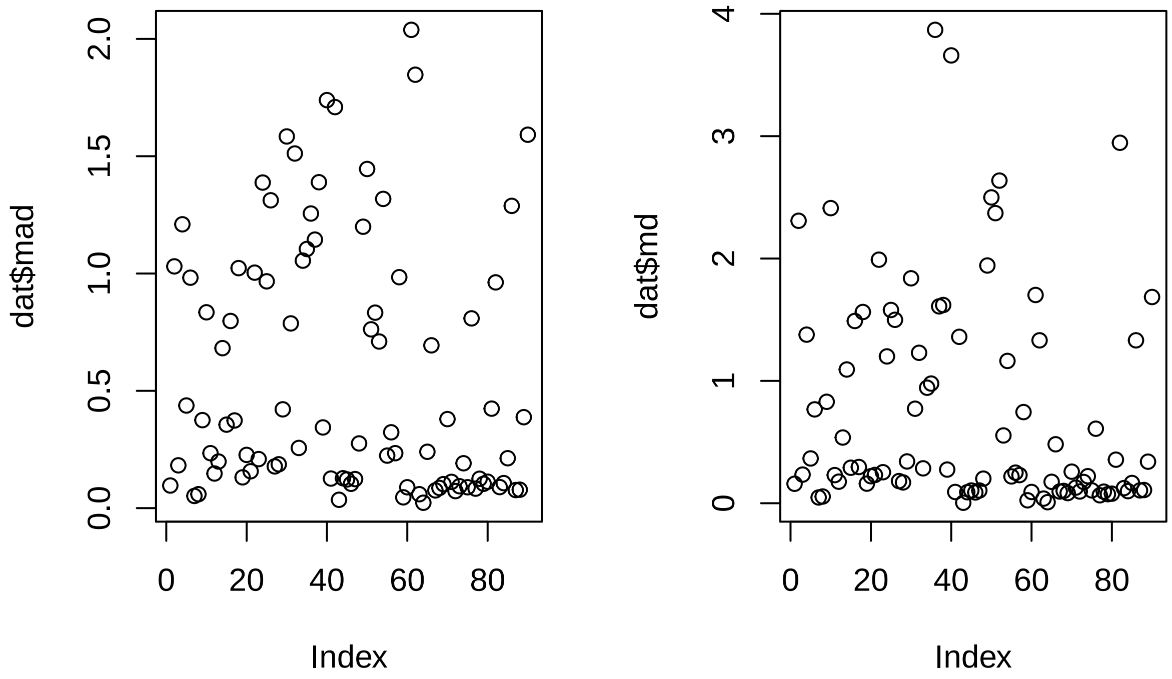
I think first and foremost it's the relative MAD we should focus on. That's the equivalent of CV (Coefficient of Variation).

```
dat$rMAD <- dat$mad/dat$md
summary(dat$rMAD)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3160  0.7399  1.0207  1.0869  1.2554  8.8956
```

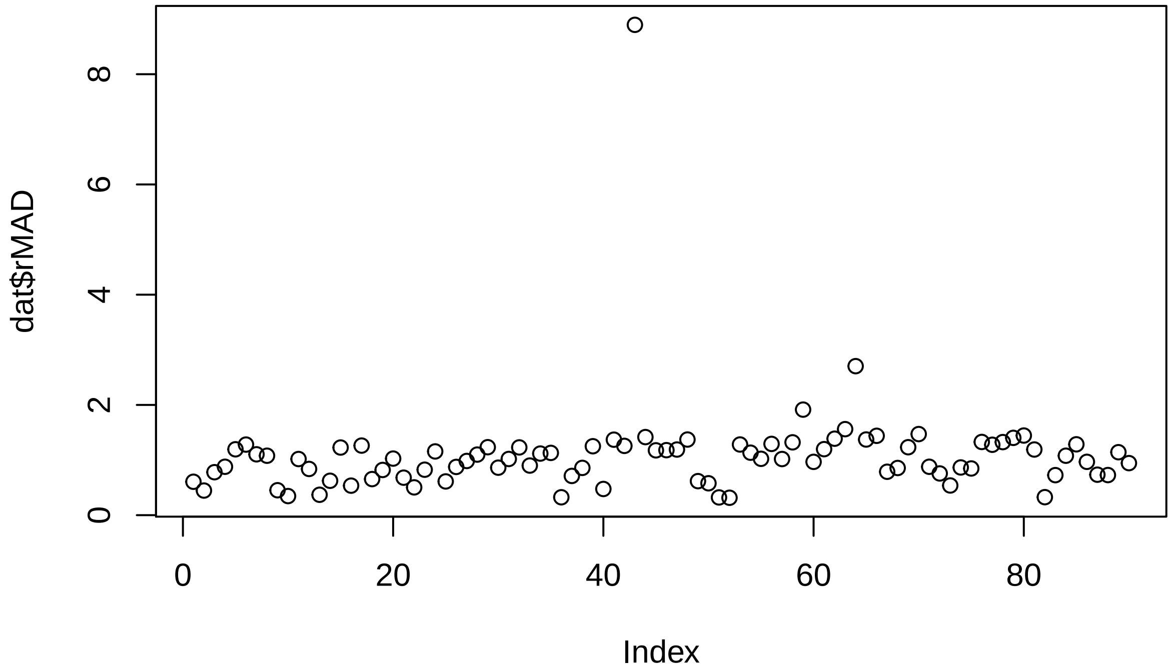
Looks sqewed.

```
par(mfrow=c(1,2))
plot(dat$mad)
plot(dat$md)
```



The two factors are well-behaved at first glance.

```r
plot(dat$rMAD)
```



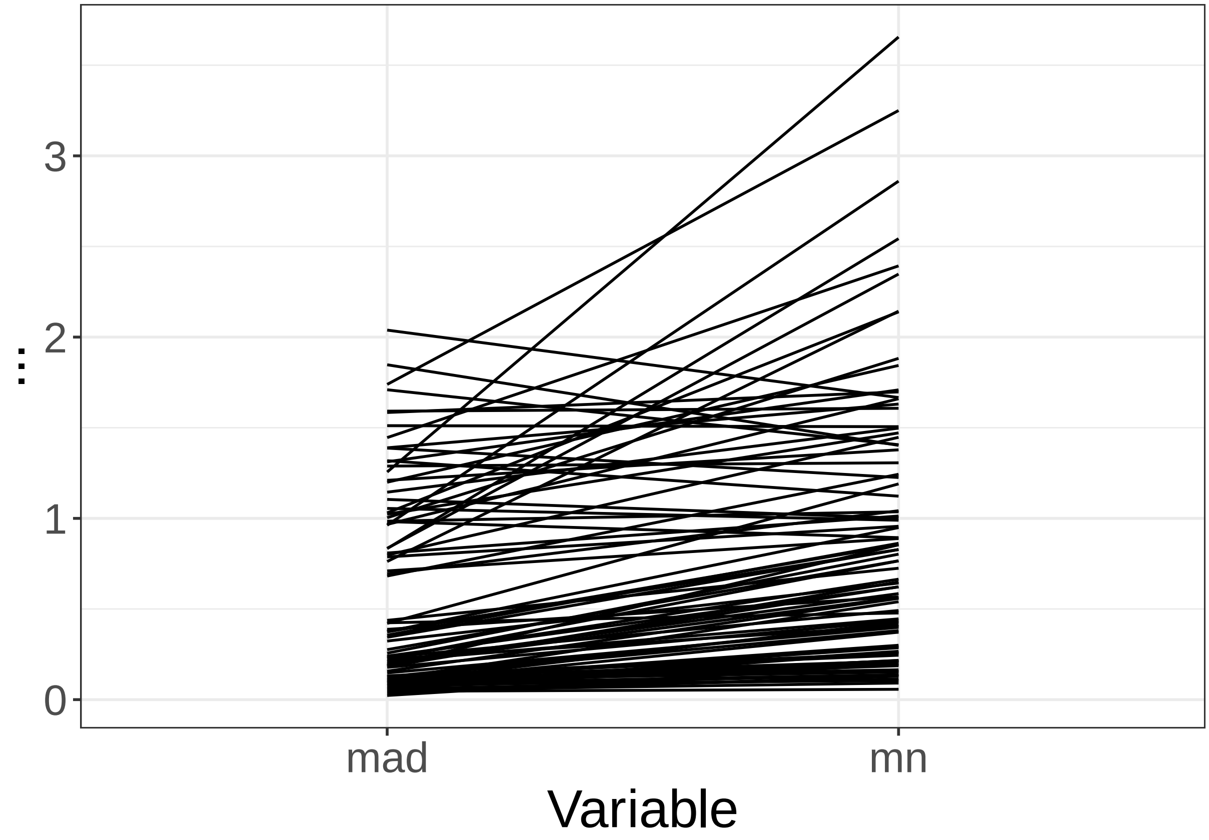
But this one is not.

```
dat[dat$rMAD>4, c("LocalityName", "Treatment", "md", "mn", "sd", "mad", "rMAD")]
```

```
## # A tibble: 1 x 7
##   LocalityName    Treatment      md    mn    sd    mad  rMAD
##   <chr>           <fct>       <dbl> <dbl> <dbl>  <dbl> <dbl>
## 1 namdalseid_1kub Open plot 0.00400 0.129 0.688 0.0356  8.90
```

It's the median that is causing trubble. It's very low, 0.004m.

```
library(reshape2)
myMelt <- melt(data = dat, id.vars = "LocalityCode", measure.vars = c("mad", "mn"))
ggplot(myMelt, aes(x=variable, y=value, group=LocalityCode))+
  geom_line()+
  labs(y='...', x='Variable')+
  theme_bw()+
  theme(text = element_text(size = 20))
```



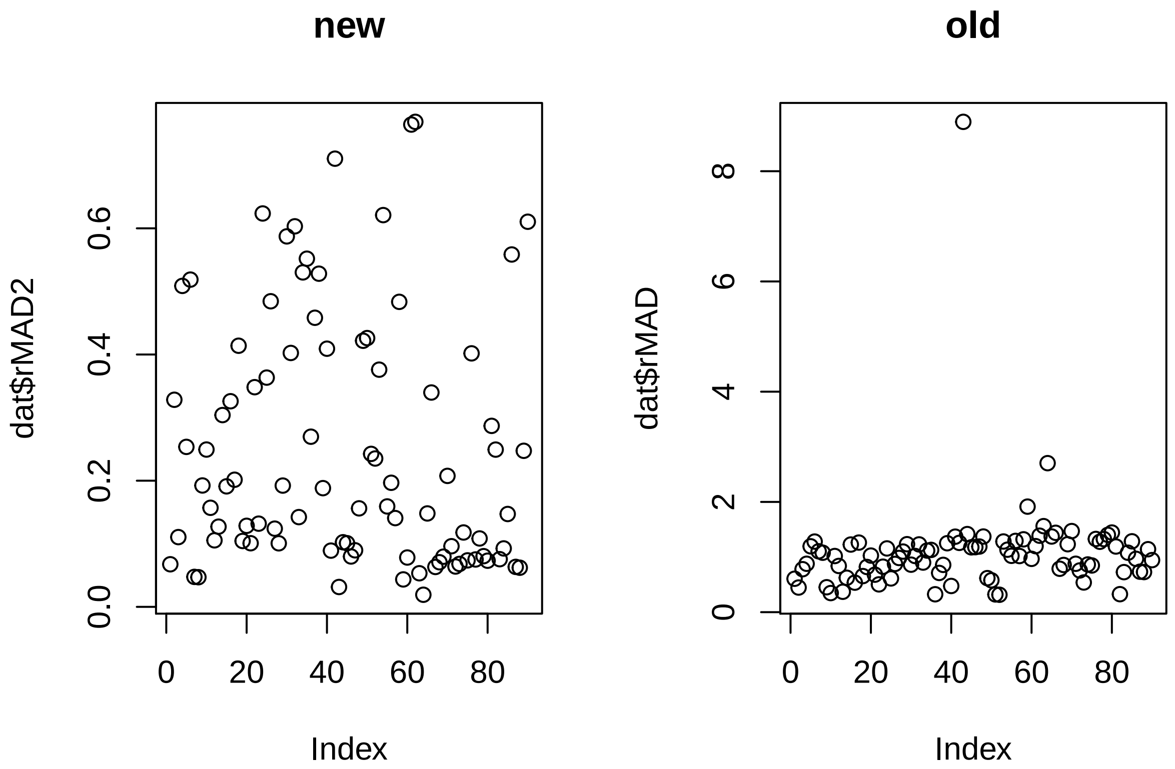
This doesn't look strange. Is is simply the low median value that increases rMAD asymptotically. It's of coarse the lowers median in the data set, as you can see here:

```
min(dat$md)
```

```
## [1] 0.004
```

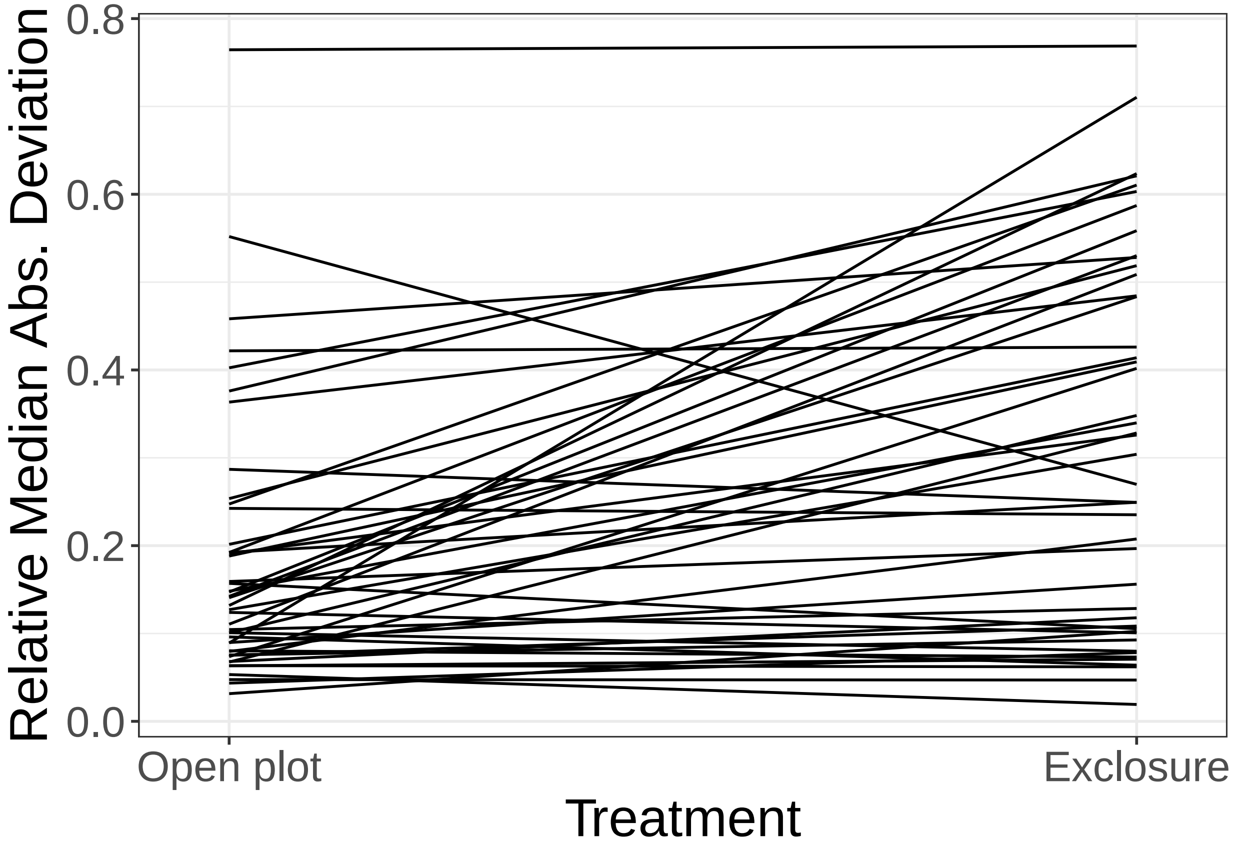
Lets remake rMAD with a slight moderation:

```
dat$rMAD2 <- dat$mad/(dat$mn+1)
par(mfrow=c(1,2))
plot(dat$rMAD2, main = "new")
plot(dat$rMAD, main = "old")
```

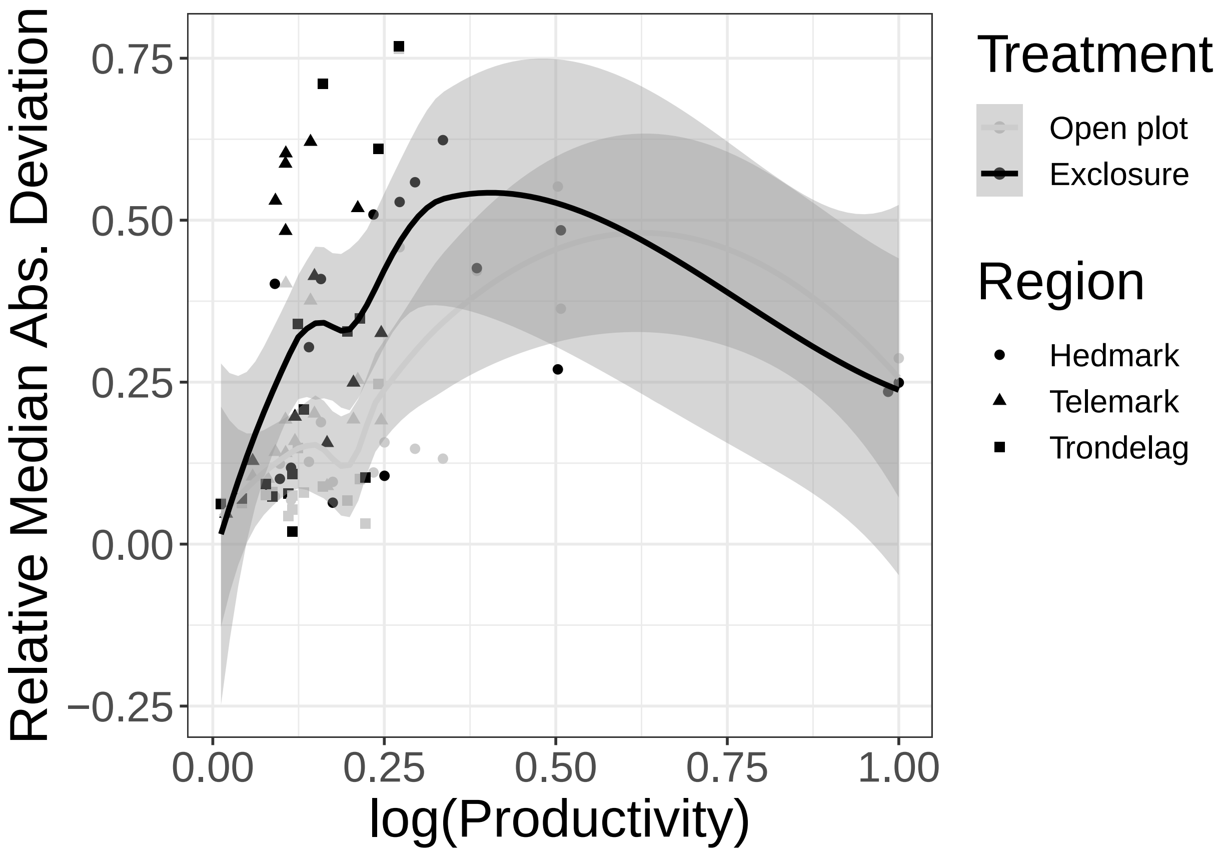


I don't see how this can affect the interpretation of rMAD. Let's use it.

```
ggplot(dat, aes(x=Treatment, y=rMAD2, group=LocalityName))+
  geom_line()+
  labs(y='Relative Median Abs. Deviation', x='Treatment')+
  scale_linetype_manual(breaks = c("Exclosure", "Open plot"),
                        labels = c("Open plots", "Exclosures"), values=c(1,2))+
  scale_x_discrete(limits = c('Open plot', 'Exclosure'),
                   breaks = c('Open plot', 'Exclosure'), expand = c(0.1,0))+
  theme_bw()+
  theme(text = element_text(size = 20))
```

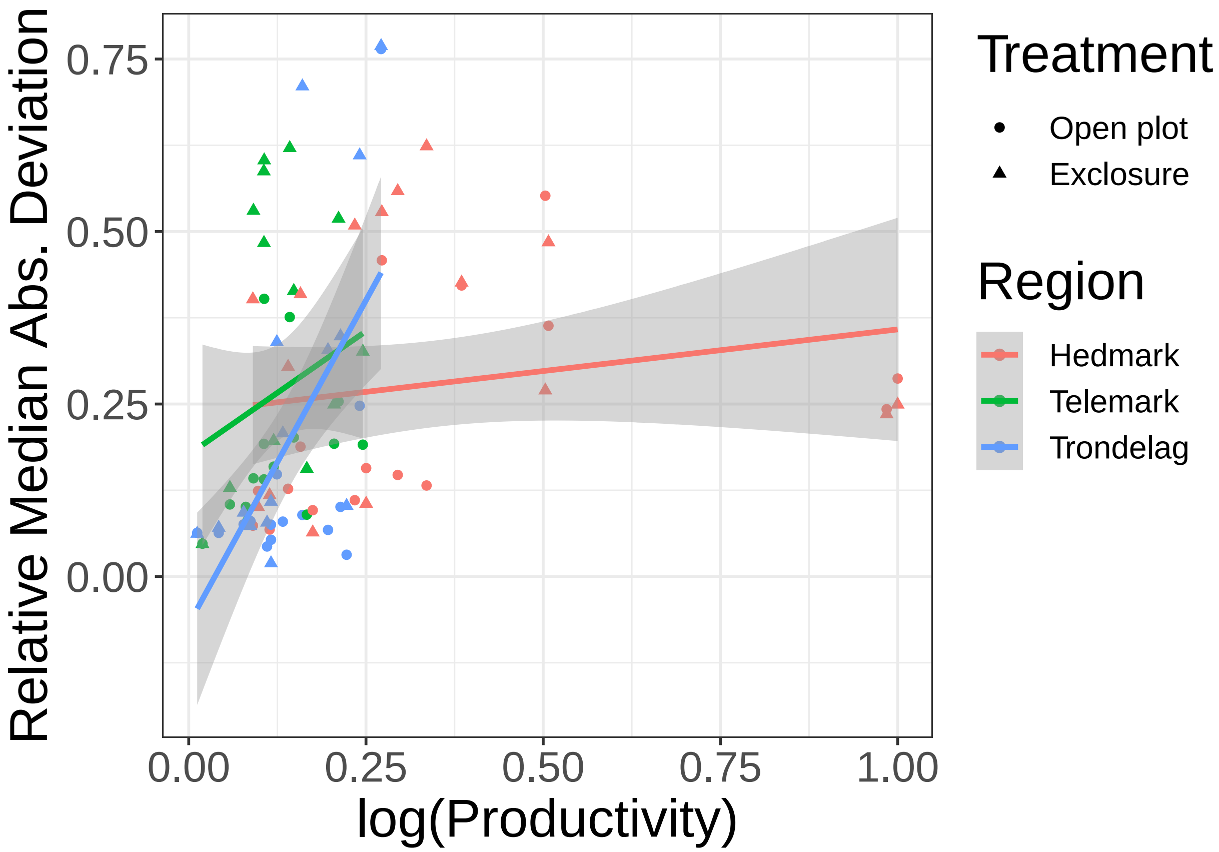


```
ggplot(data = dat,
                    aes(x = prod, y = rMAD2))+
  geom_point(aes(colour= Treatment, shape=region))+
   geom_smooth(aes(colour= Treatment),
               method = "loess", formula =  ’y ~ x’)+
  labs(y=’Relative Median Abs. Deviation’, x=’log(Productivity)’)+
  theme_bw()+
  scale_color_manual(values = c("gray80", "black"))+
  labs(colour="Treatment", shape="Region")+
  theme(text = element_text(size = 20))+
  #ylim(0, 0.4)+
  theme(legend.position = ’right’,
                             legend.justification = c("left", "top"),
                             legend.box.just = "left",
                             #legend.margin = margin(5, 5, 5, 5),
                             legend.text = element_text(size=12))
```



Prorbably no interaction with productivityhere, but the 'exclosure line' appear to lie higher. Let's see if there an effect of region

```
ggplot(data = dat,
                    aes(x = prod, y = rMAD2))+
  geom_point(aes(colour=region, shape=Treatment))+
   geom_smooth(aes(colour= region),
               method = "lm", formula =  'y ~ x')+
  labs(y='Relative Median Abs. Deviation', x='log(Productivity)')+
  theme_bw()+
  #scale_color_manual(values = c("gray80", "black", "grey20"))+
  labs(colour="Region", shape="Treatment")+
  theme(text = element_text(size = 20))+
  #ylim(0, 0.4)+
  theme(legend.position = 'right',
                              legend.justification = c("left", "top"),
                              legend.box.just = "left",
                              #legend.margin = margin(5, 5, 5, 5),
                              legend.text = element_text(size=12))
```



Looks like we need a random slope for this one.

```
modmad <- glmmTMB(data = dat,
                  rMAD2~Treatment*prod_l+prod_l2+(prod_l|region),
                  family = gaussian)
```

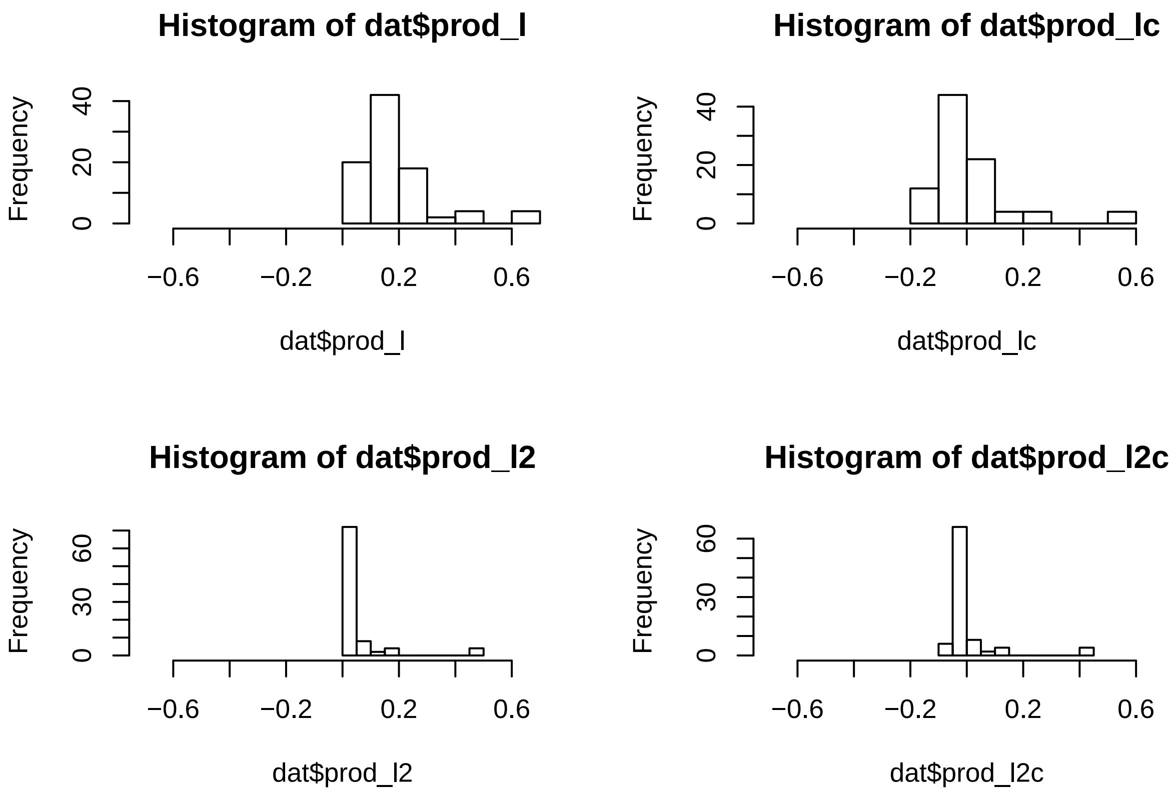
```
## Warning in fitTMB(TMBStruc): Model convergence problem; singular convergence
## (7). See vignette('troubleshooting')
```

```
summary(modmad)
```

```
##  Family: gaussian  ( identity )
## Formula:          rMAD2 ~ Treatment * prod_l + prod_l2 + (prod_l | region)
## Data: dat
##
##      AIC      BIC   logLik deviance df.resid
##    -60.3    -37.8     39.1    -78.3       81
##
## Random effects:
##
## Conditional model:
##  Groups   Name        Variance Std.Dev. Corr
##  region   (Intercept) 0.003101 0.05569
##           prod_l      0.026468 0.16269  -1.00
##  Residual             0.023795 0.15426
## Number of obs: 90, groups:  region, 3
##
## Dispersion estimate for gaussian family (sigma^2): 0.0238
##
## Conditional model:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -0.07869    0.06385  -1.232  0.21779
## TreatmentExclosure          0.16853    0.05301   3.179  0.00148 **
## prod_l                      2.14520    0.42564   5.040 4.66e-07 ***
## prod_l2                    -2.51124    0.58296  -4.308 1.65e-05 ***
## TreatmentExclosure:prod_l -0.18024    0.23236  -0.776  0.43792
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note perfect correlation between random effect. All I can think of trying the centering.

```r
dat$prod_lc <- dat$prod_l-mean(dat$prod_l)
dat$prod_l2c <- dat$prod_l2-mean(dat$prod_l2)
par(mfrow=c(2,2))
hist(dat$prod_l, xlim = c(-0.7,0.7))
hist(dat$prod_lc, xlim = c(-0.7,0.7))
hist(dat$prod_l2, xlim = c(-0.7,0.7))
hist(dat$prod_l2c, xlim = c(-0.7,0.7))
```



Ok, but notice how sqewed the quadratic term is, probably amplifying the 'outlierissue'…

```
modmad2 <- glmmTMB(data = dat,
                  rMAD2~Treatment*prod_lc+prod_l2c+(prod_lc|region),
                  family = gaussian)
```

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

## Warning in fitTMB(TMBStruc): Model convergence problem; singular convergence
## (7). See vignette('troubleshooting')
```

```
summary(modmad2)
```

```
##  Family: gaussian  ( identity )
## Formula:          rMAD2 ~ Treatment * prod_lc + prod_l2c + (prod_lc | region)
## Data: dat
##
##      AIC      BIC   logLik deviance df.resid
##       NA       NA       NA       NA       81
##
## Random effects:
##
## Conditional model:
##  Groups   Name        Variance Std.Dev. Corr
##  region   (Intercept) 0.000696 0.02638
##           prod_lc     0.026468 0.16269  -1.00
##  Residual             0.023795 0.15426
## Number of obs: 90, groups:  region, 3
##
## Dispersion estimate for gaussian family (sigma^2): 0.0238
##
## Conditional model:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  0.17708    0.02835   6.247 4.19e-10 ***
## TreatmentExclosure           0.13606    0.03252   4.184 2.87e-05 ***
## prod_lc                      2.14520    0.42564   5.040 4.66e-07 ***
## prod_l2c                    -2.51124    0.58296  -4.308 1.65e-05 ***
## TreatmentExclosure:prod_lc -0.18024    0.23236  -0.776    0.438
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Well that just made it worse. I think we just need to simplyfy the model.

```
modmad3 <- glmmTMB(data = dat,
                  rMAD2~Treatment*prod_l+prod_l2+(1|region),
                  family = gaussian)
summary(modmad3)
```

```
##  Family: gaussian  ( identity )
## Formula:          rMAD2 ~ Treatment * prod_l + prod_l2 + (1 | region)
## Data: dat
##
```

```
##      AIC      BIC   logLik deviance df.resid
##    -63.3    -45.8     38.7    -77.3       83
##
## Random effects:
##
## Conditional model:
##  Groups   Name        Variance  Std.Dev.
##  region   (Intercept) 0.0008213 0.02866
##  Residual             0.0242256 0.15565
## Number of obs: 90, groups:  region, 3
##
## Dispersion estimate for gaussian family (sigma^2): 0.0242
##
## Conditional model:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -0.07861    0.05777  -1.361  0.17358
## TreatmentExclosure         0.16853    0.05348   3.151  0.00163 **
## prod_l                     2.14061    0.41378   5.173 2.30e-07 ***
## prod_l2                   -2.41072    0.53976  -4.466 7.96e-06 ***
## TreatmentExclosure:prod_l -0.18024    0.23445  -0.769  0.44202
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

Like so. Lets for fun compare with the random slope model, but I’m not sure if its AIC is compareable/reliable:

```r
AIC(modmad, modmad3)
```

```
##         df       AIC
## modmad   9 -60.25425
## modmad3  7 -63.31825
```

The random intecept model is better anyways. Byt we can drop the interaction, and theat leave us with:

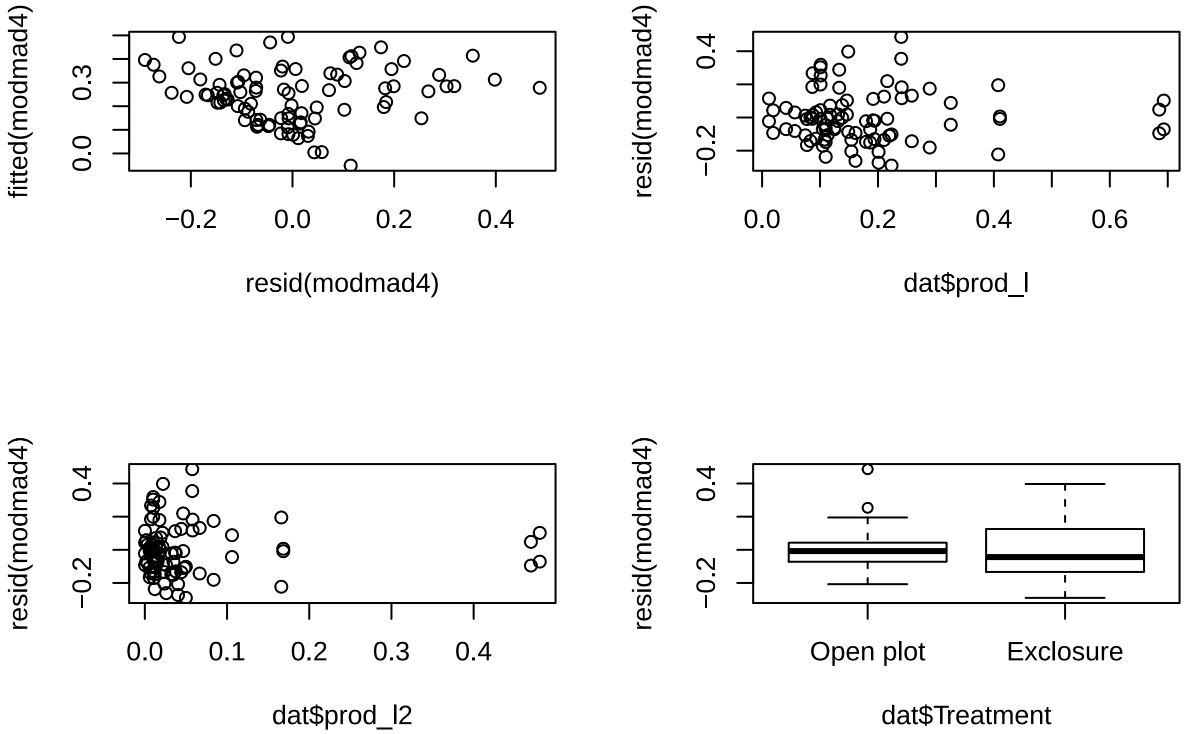
```r
modmad4 <- update(modmad3, .~. -Treatment:prod_l)
summary(modmad4)
```

```
##  Family: gaussian  ( identity )
## Formula:          rMAD2 ~ Treatment + prod_l + prod_l2 + (1 | region)
## Data: dat
##
##      AIC      BIC   logLik deviance df.resid
##    -64.7    -49.7     38.4    -76.7       84
##
## Random effects:
##
## Conditional model:
##  Groups   Name        Variance  Std.Dev.
##  region   (Intercept) 0.0008136 0.02852
##  Residual             0.0243913 0.15618
## Number of obs: 90, groups:  region, 3
##
```

```
## Dispersion estimate for gaussian family (sigma^2): 0.0244
##
## Conditional model:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.06226    0.05391  -1.155    0.248
## TreatmentExclosure   0.13606    0.03293   4.132 3.59e-05 ***
## prod_l               2.04957    0.39815   5.148 2.64e-07 ***
## prod_l2             -2.40999    0.54160  -4.450 8.60e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All very significant. Error estimates seem reasonable. Let's look at some validation plots.

```r
par(mfrow = c(2,2))
plot(resid(modmad4), fitted(modmad4))
plot(dat$prod_l, resid(modmad4))
plot(dat$prod_l2, resid(modmad4))
plot(resid(modmad4)~dat$Treatment)
```

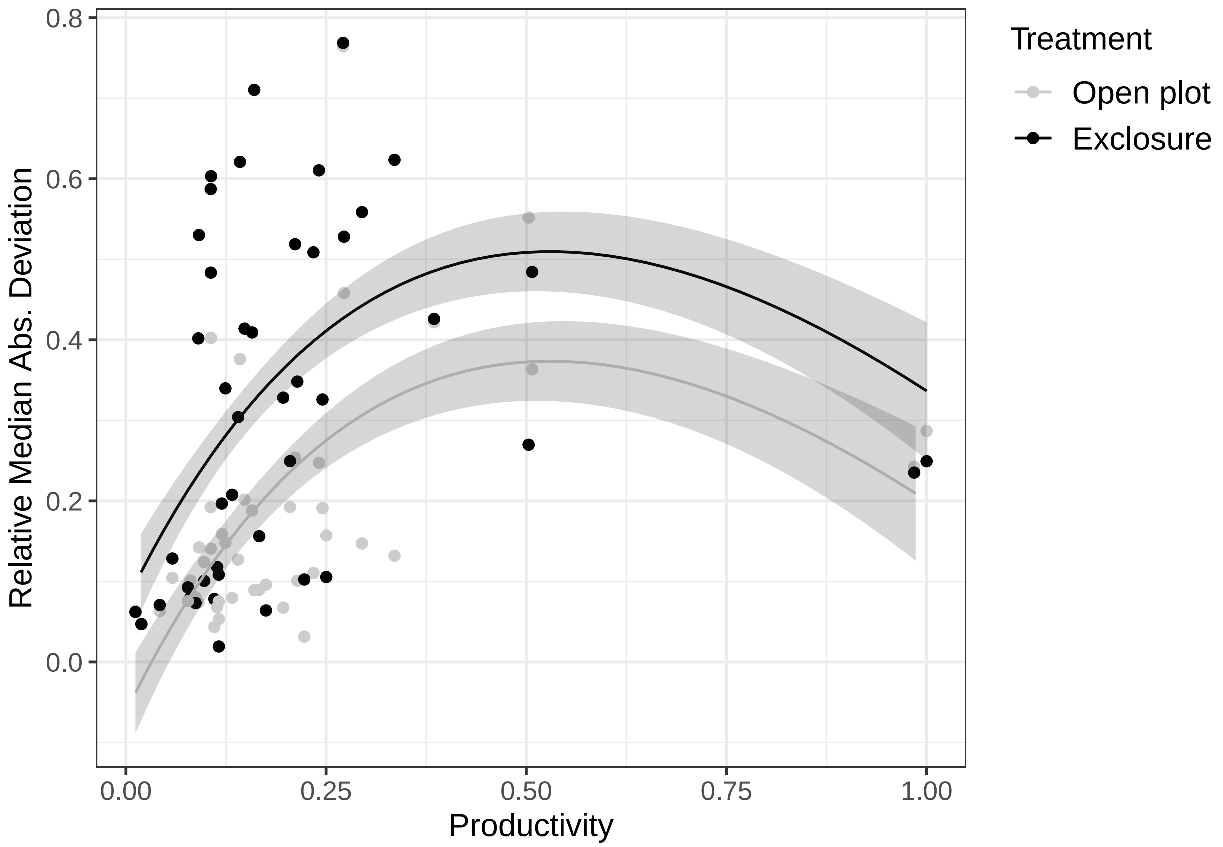


Looks fine.

Plot rMAD2

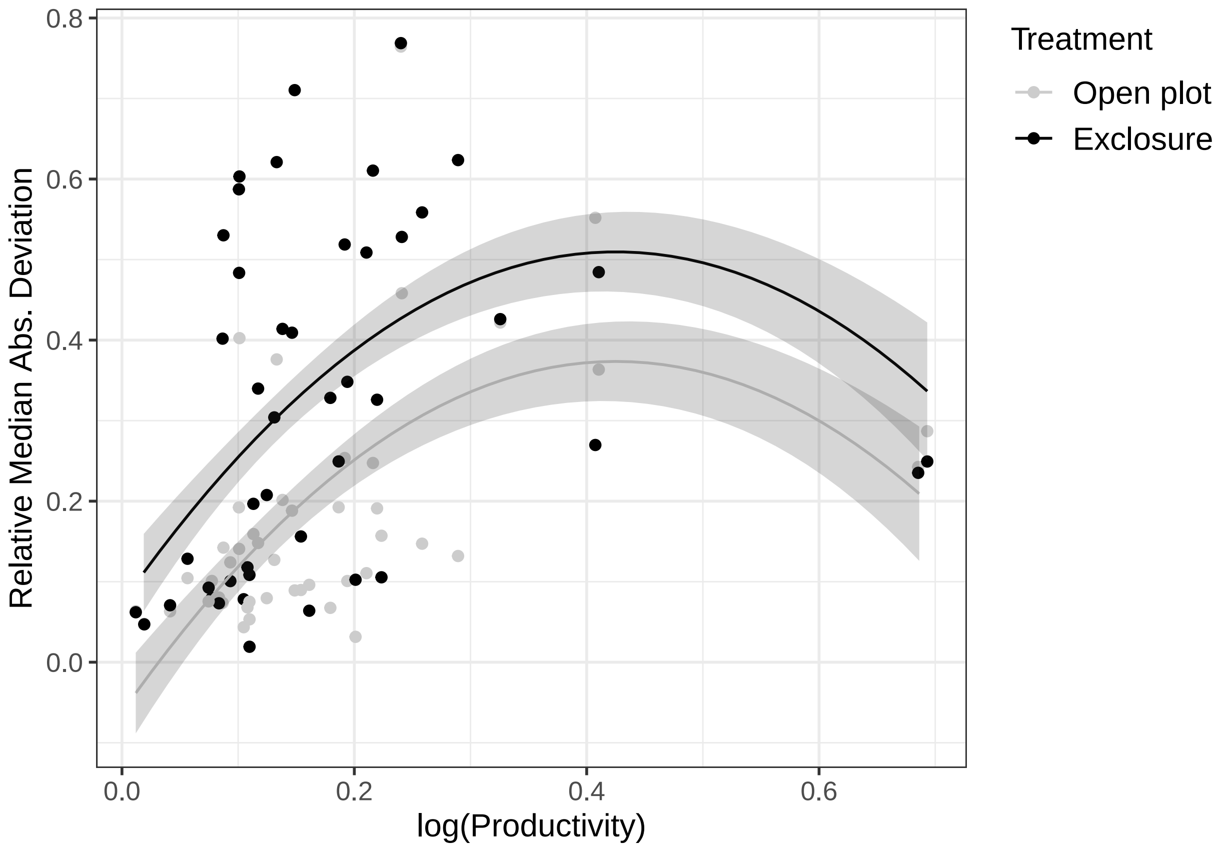
```
tr <- rep(c("Open plot", "Exclosure"), times=50, each=1)
prod_l <- seq(from = min(dat$prod_l), to=max(dat$prod_l), length.out = 100)
prod_l2 <- prod_l^2
region <- rep(NA, times=100)

pred <- predict(modmad4, list(Treatment=tr,
                              prod_l=prod_l,
                              prod_l2=prod_l2,
                              region = region), se.fit = TRUE)
pred2 <- data.frame(Treatment = tr,
                    prod_l    = prod_l,
                    pred      = pred$fit,
                    se        = pred$se.fit)
```

```
ggplot()+
  geom_point(data = dat,aes(x = exp(prod_l)-1, y = rMAD2, colour= Treatment))+
  geom_line(data=pred2, aes(x = exp(prod_l)-1, y=pred, colour = Treatment))+
  geom_ribbon(data=pred2, aes(x = exp(prod_l)-1,
                              ymin=pred-se,
                              ymax=pred+se,
                              group = Treatment),
                              alpha=0.2,
                              linetype="blank")+
  labs(y="Relative Median Abs. Deviation", x=’Productivity’)+
  theme_bw()+
  scale_color_manual(values = c("gray80", "black"))+
  labs(colour="Treatment")+
  theme(text = element_text(size = 12))+
  theme(legend.position = ’right’,
                             legend.justification = c("left", "top"),
                             legend.box.just = "left",
                             #legend.margin = margin(5, 5, 5, 5),
                             legend.text = element_text(size=12))
```



That looks a bot odd. Lets look at it without backtransforming the x-axis:

```
ggplot()+
  geom_point(data = dat,aes(x = prod_l, y = rMAD2, colour= Treatment))+
  geom_line(data=pred2, aes(x = prod_l, y=pred, colour = Treatment))+
  geom_ribbon(data=pred2, aes(x = prod_l,
                              ymin=pred-se,
                              ymax=pred+se,
                              group = Treatment),
                              alpha=0.2,
                              linetype="blank")+
  labs(y="Relative Median Abs. Deviation", x='log(Productivity)')+
  theme_bw()+
  scale_color_manual(values = c("gray80", "black"))+
  labs(colour="Treatment")+
  theme(text = element_text(size = 12))+
  theme(legend.position = 'right',
                             legend.justification = c("left", "top"),
                             legend.box.just = "left",
                             #legend.margin = margin(5, 5, 5, 5),
                             legend.text = element_text(size=12))
```



Still strange.

Sensitivity analysis

Removing the most productive sites and comparing the results

```
dat2 <- dat[dat$prod<0.7,]
dat2$prod2 <- dat2$prod^2
```

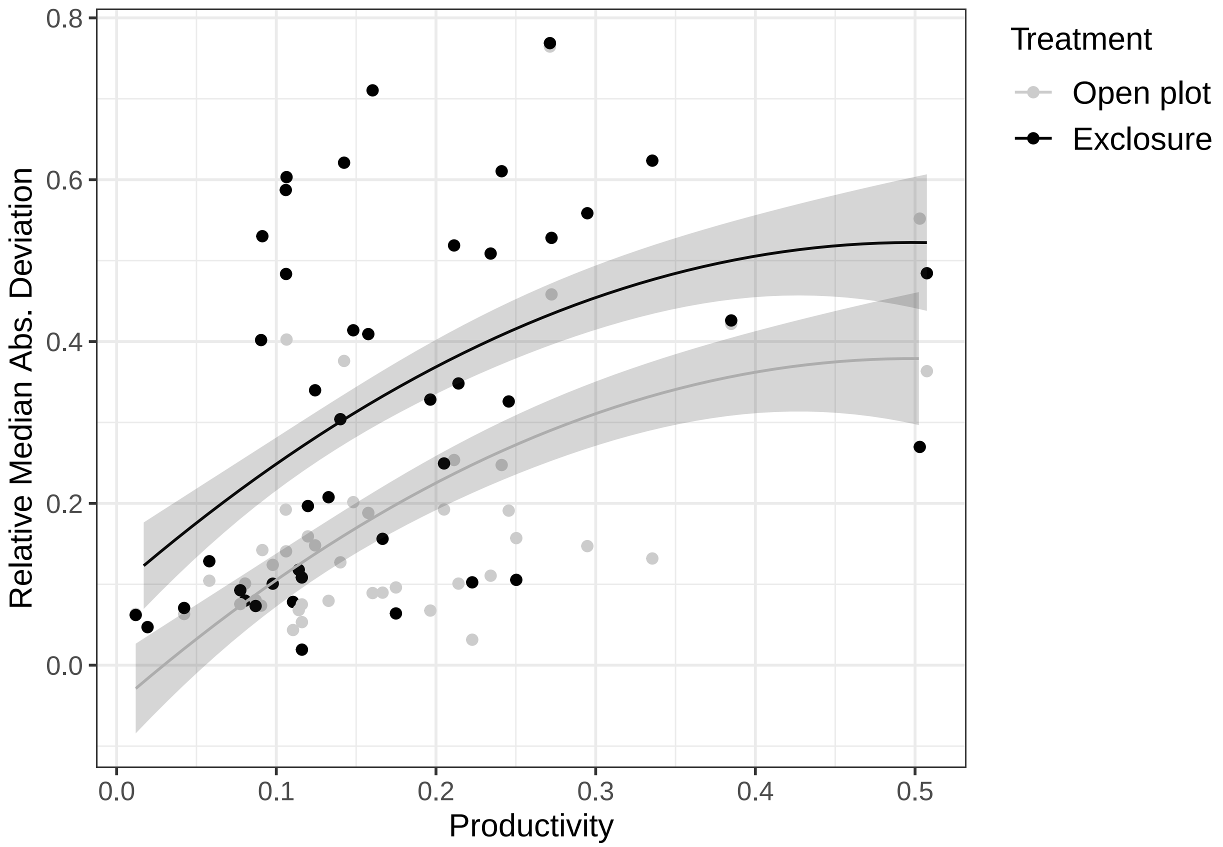
```
mads <- glmmTMB(data=dat2,
                rMAD2~Treatment*prod+prod2+(1|region), family = gaussian)
mads2 <- update(mads, .~. -prod2)
mads3 <- update(mads2, .~. -Treatment:prod)
mads4 <- update(mads, .~. -Treatment:prod)

AIC(mads, mads2, mads3, mads4)
```

```
##       df       AIC
## mads   7 -56.70222
## mads2  6 -55.74227
## mads3  5 -57.73926
## mads4  6 -58.69911
```

The interaction term is dropped when not including the most productive sites.

```r
tr <- rep(c("Open plot", "Exclosure"), times=50, each=1)
prod <- seq(from = min(dat2$prod), to=max(dat2$prod), length.out = 100)
prod2 <- prod^2
region <- rep(NA, times=100)
pred <- predict(mads4, list(Treatment=tr,
                              prod   =prod,
                              prod2  =prod2,
                              region =region), se.fit = TRUE)
pred2 <- data.frame(Treatment = tr,
                    prod      = prod,
                    pred      = pred$fit,
                    se        = pred$se.fit)
```

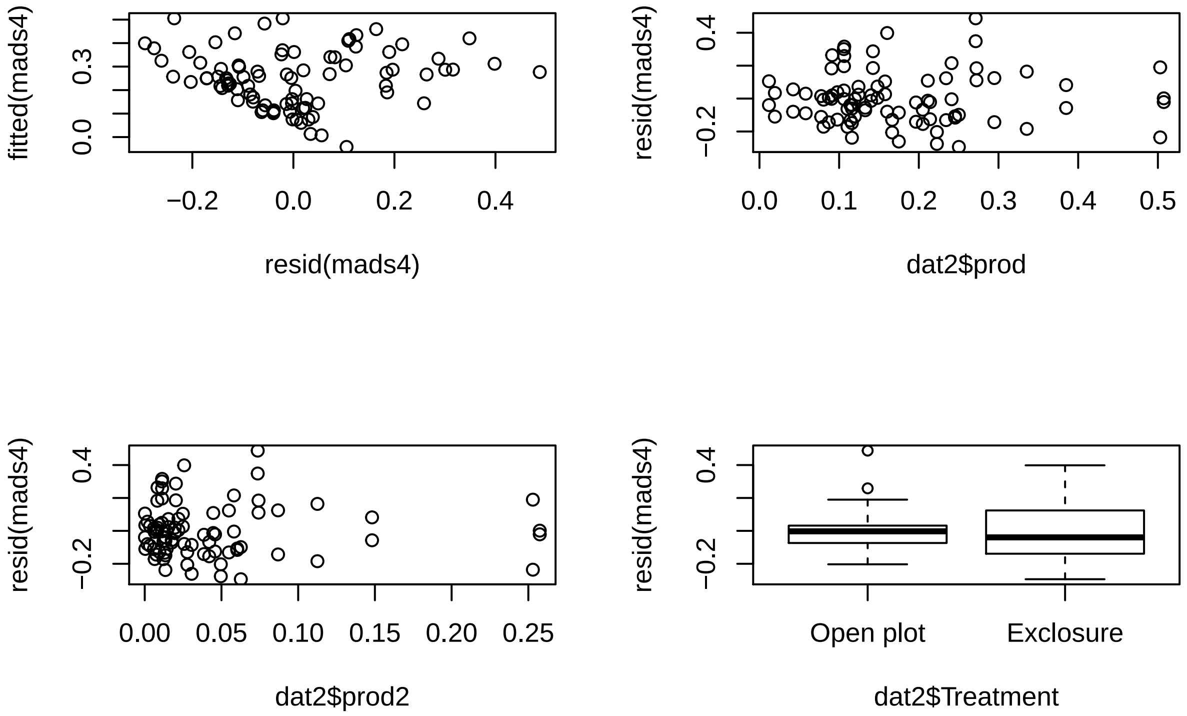
```r
ggplot()+
  geom_point(data = dat2,aes(x = prod, y = rMAD2, colour= Treatment))+
  geom_line(data=pred2, aes(x = prod, y=pred, colour = Treatment))+
  geom_ribbon(data=pred2, aes(x = prod,
                              ymin=pred-se,
                              ymax=pred+se,
                              group = Treatment),
                              alpha=0.2,
                              linetype="blank")+
  labs(y="Relative Median Abs. Deviation", x=’Productivity’)+
  theme_bw()+
  scale_color_manual(values = c("gray80", "black"))+
  labs(colour="Treatment")+
  theme(text = element_text(size = 12))+
  theme(legend.position = ’right’,
                             legend.justification = c("left", "top"),
                             legend.box.just = "left",
                             #legend.margin = margin(5, 5, 5, 5),
                             legend.text = element_text(size=12))
```



The plots are very different looking, but from 0 - 0.5 the lines follow a quite similar path. Probably this last one is more logical, as there no reason to think that rMAD should go towards zero as prod increases. It's however a little confusing for the reader if we sometimes take the points out and sometimes not… That makes me opt for removing them from the start. We can include the plots as appendix as a safeguard perhaps.

Lets validata that last model.

```
par(mfrow=c(2,2))
plot(resid(mads4), fitted(mads4))
plot(dat2$prod, resid(mads4))
plot(dat2$prod2, resid(mads4))
plot(resid(mads4)~dat2$Treatment)
```



ok.

```
summary(mads4)
```

```
##  Family: gaussian  ( identity )
## Formula:          rMAD2 ~ Treatment + prod + prod2 + (1 | region)
## Data: dat2
##
##      AIC      BIC   logLik deviance df.resid
##    -58.7    -44.0     35.3    -70.7       80
##
## Random effects:
##
## Conditional model:
##  Groups   Name        Variance Std.Dev.
##  region   (Intercept) 0.00089  0.02983
##  Residual             0.02511  0.15847
## Number of obs: 86, groups:  region, 3
##
## Dispersion estimate for gaussian family (sigma^2): 0.0251
##
## Conditional model:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.04907    0.06038  -0.813 0.416356
## TreatmentExclosure   0.14343    0.03418   4.197 2.71e-05 ***
## prod                 1.71534    0.51583   3.325 0.000883 ***
## prod2               -1.71848    0.99005  -1.736 0.082607 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So the interaction isn't significant, but lowers the AIC none the less. Summary stats:

```
m <- tapply(dat2$rMAD2, dat2$Treatment, FUN = mean)
# And standard error:
source('se.R')
s <- tapply(dat2$rMAD2, dat2$Treatment, FUN = se)
data.frame("mean canopy growth per year" = round(m, 3),
                        "standard error of the mean" = round(s, 3))
```

```
##           mean.canopy.growth.per.year standard.error.of.the.mean
## Open plot                       0.177                      0.023
## Exclosure                       0.320                      0.033
```

Canopy growth v2

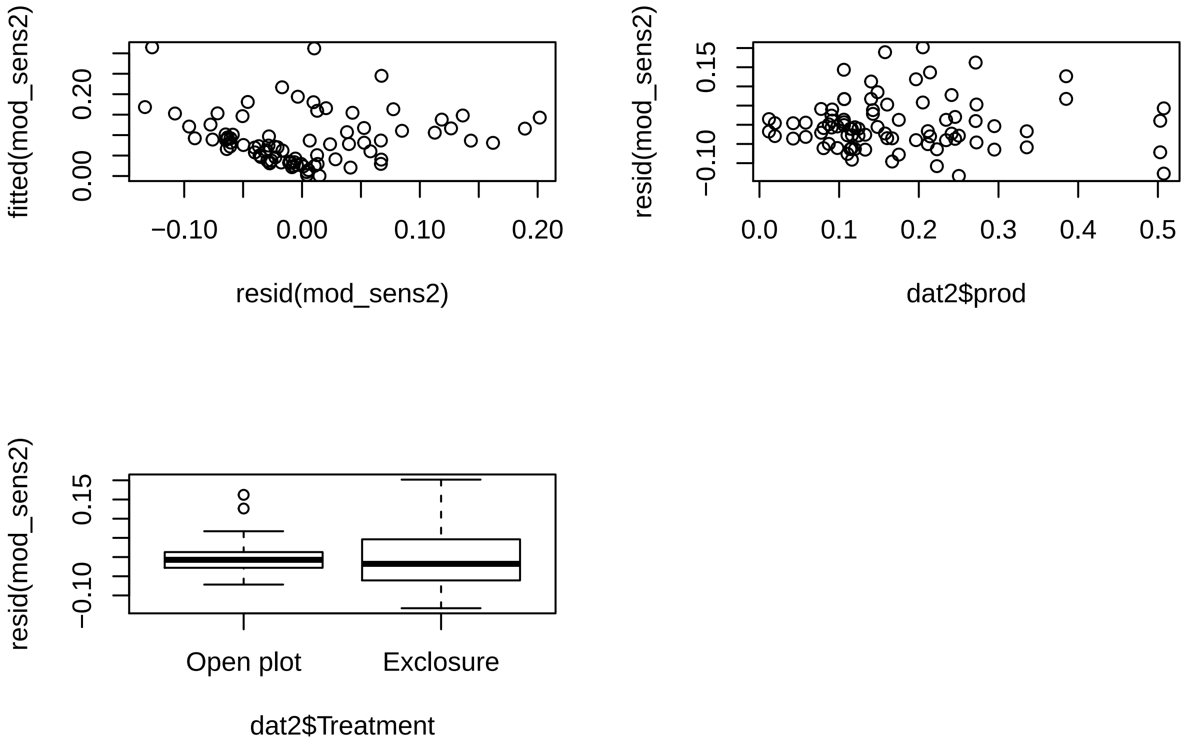
Because we decided to drop the 'outliers', we want to da that also in the canopy growth analysis. We have the model:

```
summary(mod_sens2)
```

```
##  Family: gaussian  ( identity )
## Formula:          canopygrowth ~ Treatment * prod
## Data: dat2
##
##      AIC      BIC   logLik deviance df.resid
##   -213.5   -201.3    111.8   -223.5       81
##
##
## Dispersion estimate for gaussian family (sigma^2): 0.00435
##
## Conditional model:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -0.003533   0.018914  -0.187 0.851835
## TreatmentExclosure      0.029838   0.026749   1.115 0.264644
## prod                    0.311650   0.091980   3.388 0.000703 ***
## TreatmentExclosure:prod  0.256527   0.130079   1.972 0.048601 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Significant interaction, ok. Some validation.

```
par(mfrow = c(2,2))
plot(resid(mod_sens2), fitted(mod_sens2))
plot(dat2$prod, resid(mod_sens2))
plot(resid(mod_sens2)~dat2$Treatment)
```



Ok. And some new summary stats:

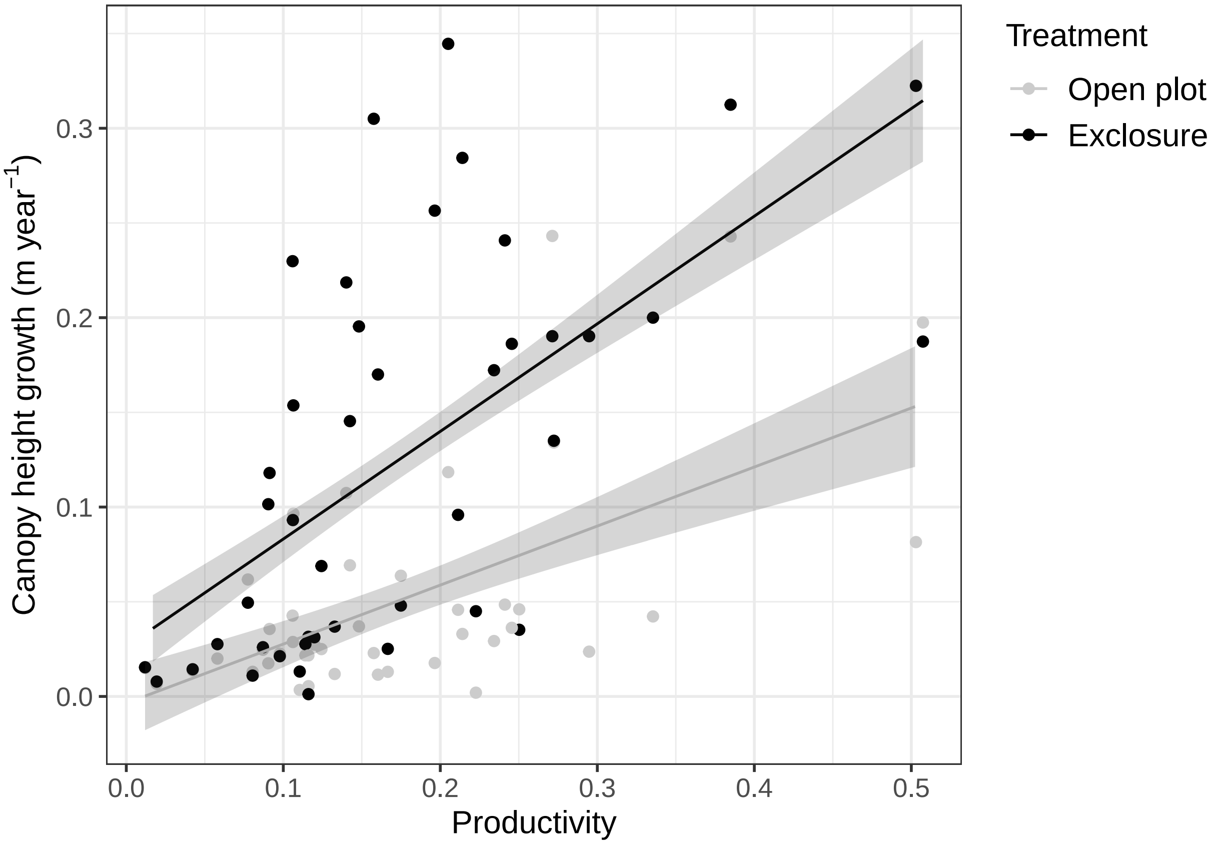
```
m <- tapply(dat2$canopygrowth, dat2$Treatment, FUN = mean)
s <- tapply(dat2$canopygrowth, dat2$Treatment, FUN = se)
data.frame("mean canopy growth per year in meters" = round(m, 3),
                        "standard error of the mean" = round(s, 3))
```

```
##           mean.canopy.growth.per.year.in.meters standard.error.of.the.mean
## Open plot                                 0.051                      0.009
## Exclosure                                 0.125                      0.016
```

And plot

```
tr <- rep(c("Open plot", "Exclosure"), times=50, each=1)
prod <- seq(from = min(dat2$prod), to=max(dat2$prod), length.out = 100)
pred <- predict(mod_sens2, list(Treatment= tr,
                                prod    = prod),
                                se.fit   = TRUE)
pred2 <- data.frame(Treatment = tr,
                    prod_l     = prod_l,
                    pred       = pred$fit,
                    se         = pred$se.fit)
```

```
ggplot()+
  geom_point(data = dat2,aes(x = prod, y = canopygrowth, colour= Treatment))+
  geom_line(data=pred2, aes(x = prod, y=pred, colour = Treatment))+
  geom_ribbon(data=pred2, aes(x = prod,
                              ymin=pred-se,
                              ymax=pred+se,
                              group = Treatment),
                              alpha=0.2,
                              linetype="blank")+
  labs(y=expression(paste(’Canopy height growth (m year’^’-1’, ’)’)), x=’Productivity’)+
  theme_bw()+
  scale_color_manual(values = c("gray80", "black"))+
  labs(colour="Treatment")+
  theme(text = element_text(size = 12))+
  theme(legend.position = ’right’,
                             legend.justification = c("left", "top"),
                             legend.box.just = "left",
                             #legend.margin = margin(5, 5, 5, 5),
                             legend.text = element_text(size=12))
```



Compare LiDAR and field data

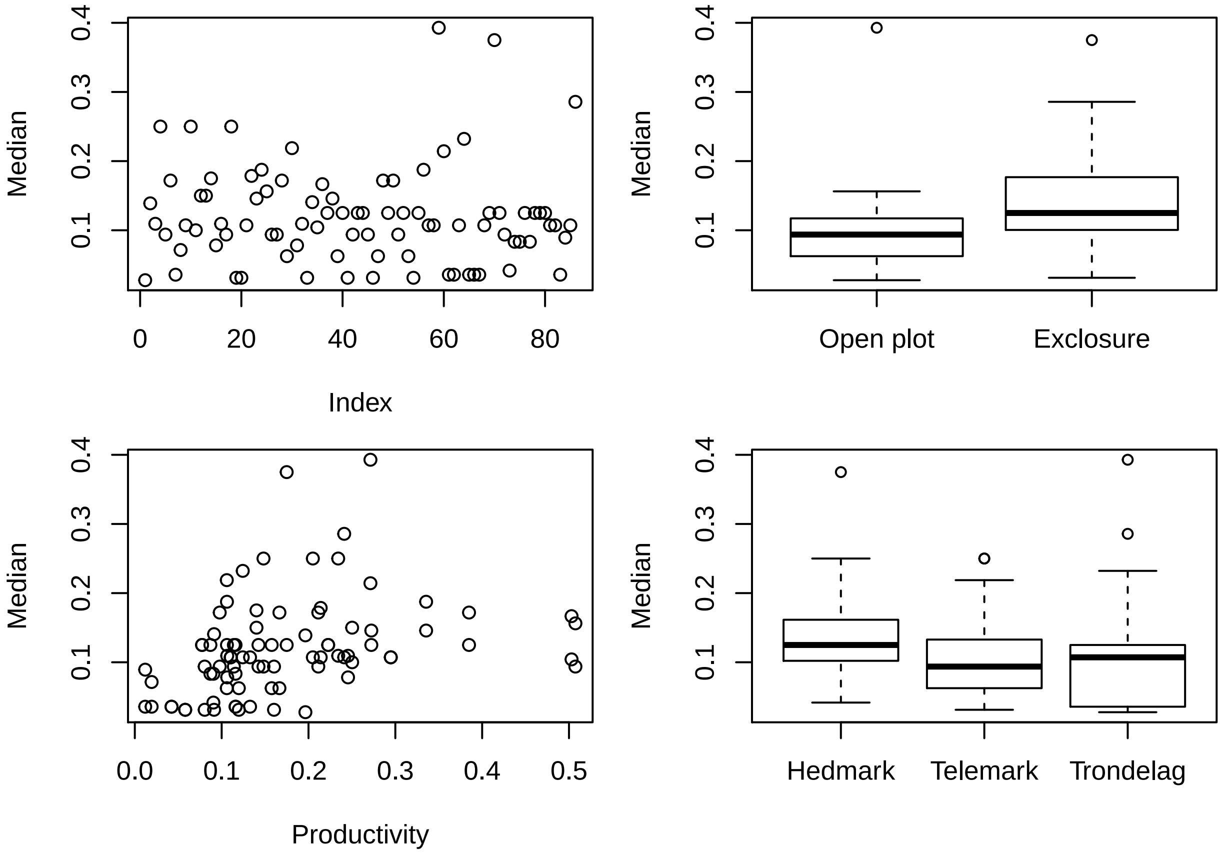
First, let’s calculate canopy growth as we did for the LiDAR data

```
dat2$canopygrowth_f <- dat2$field_median/dat2$YrsSinceExclosure
```

```
summary(dat2$canopygrowth_f)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.02778 0.07943 0.10714 0.11947 0.14583 0.39286
```

```
par(mfrow=c(2,2), mar=c(4,4,1,1))
plot(dat2$canopygrowth_f, ylab="Median")
plot(dat2$canopygrowth_f~dat2$Treatment, xlab="", ylab="Median")
plot(dat2$canopygrowth_f~dat2$prod, xlab="Productivity", ylab="Median")
plot(dat2$canopygrowth_f~dat2$region, xlab="", ylab="Median")
```



Appears the annual median tree growth of a site (median of the median of a circle) is around 0.1, or precisely:

```
median(dat2$canopygrowth_f)
```

```
## [1] 0.1071429
```

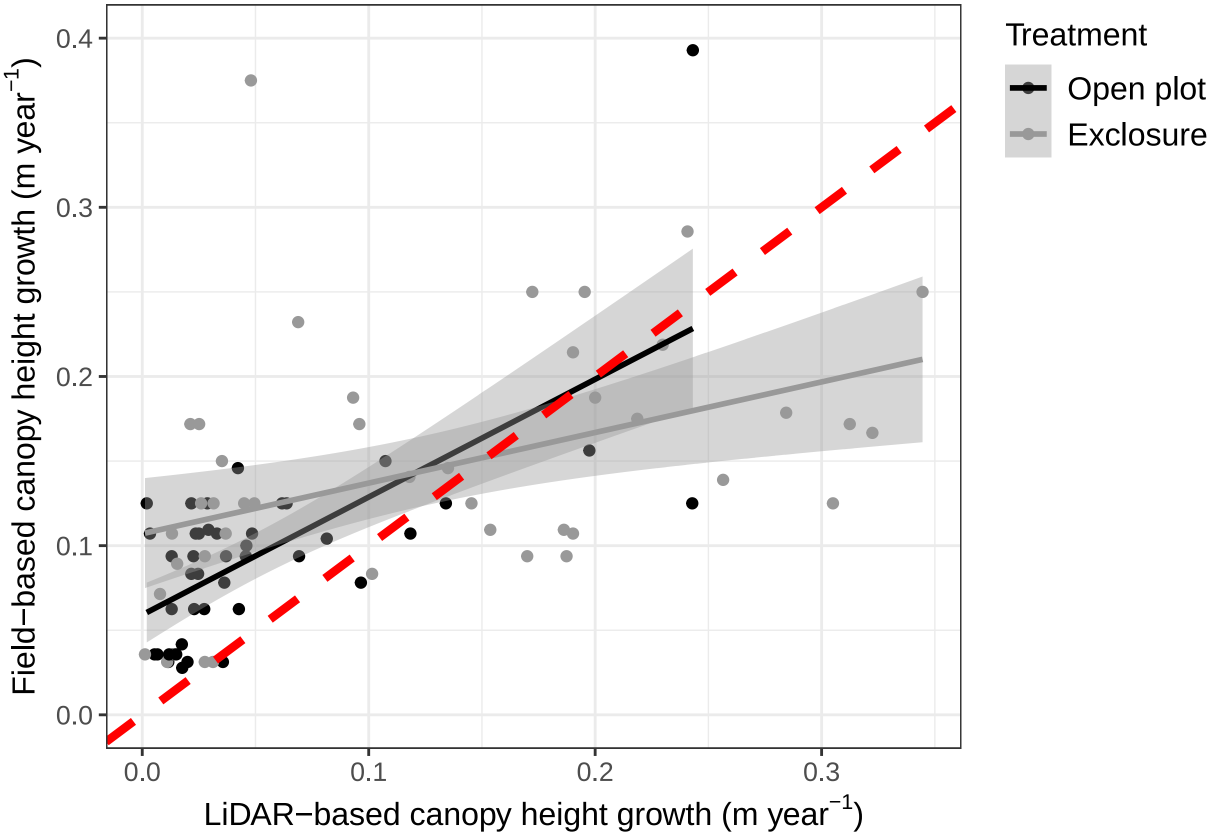
Also, it is greater in exclosures and increases with productivity, and is highest in Hedmark. Very similar to the LiDAR data.

```
m2 <- tapply(dat2$canopygrowth_f, dat2$Treatment, FUN = mean)
s2 <- tapply(dat2$canopygrowth_f, dat2$Treatment, FUN = se)
data.frame("mean canopy growth per year" = round(m2, 3),
                        "standard error of the mean" = round(s2, 3))
```

```
##            mean.canopy.growth.per.year standard.error.of.the.mean
## Open plot                        0.094                      0.009
## Exclosure                        0.145                      0.011
```

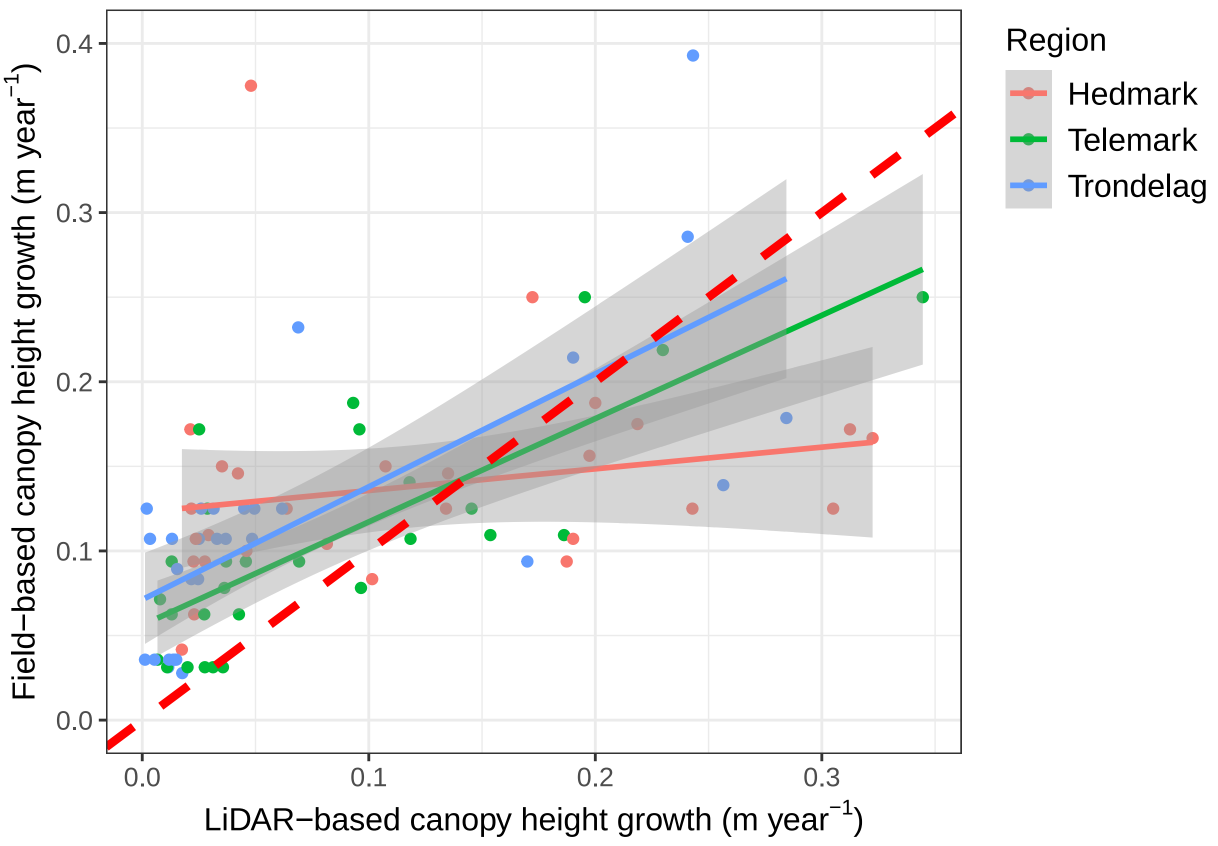
We don’t want to analyse the field data alone, or by itself. That we have done in other papers and with more data. Here we are only interested in ‘ground-truthing’ the LiDAR data. So..:

```
(corPlot <- ggplot(data = dat2,
                   aes(x = canopygrowth, y = canopygrowth_f, colour= Treatment))+
  geom_point()+
   geom_smooth(method = "lm")+
  labs(y=expression(paste(’Field-based canopy height growth (m year’^’-1’, ’)’)), x=expression(paste(’L
  theme_bw()+
  scale_color_manual(values = c("gray0", "gray60"))+
  labs(colour="Treatment")+
  theme(text = element_text(size = 12))+
  ylim(0, 0.4)+
  theme(legend.position = ’right’,
                             legend.justification = c("left", "top"),
                             legend.box.just = "left",
                             #legend.margin = margin(5, 5, 5, 5),
                             legend.text = element_text(size=12))+
   geom_abline(intercept = 0, slope = 1, color="red",
                 linetype="dashed", size=1.5)
)
```



Perhaps the correation strength differes between treatment, but not very much.

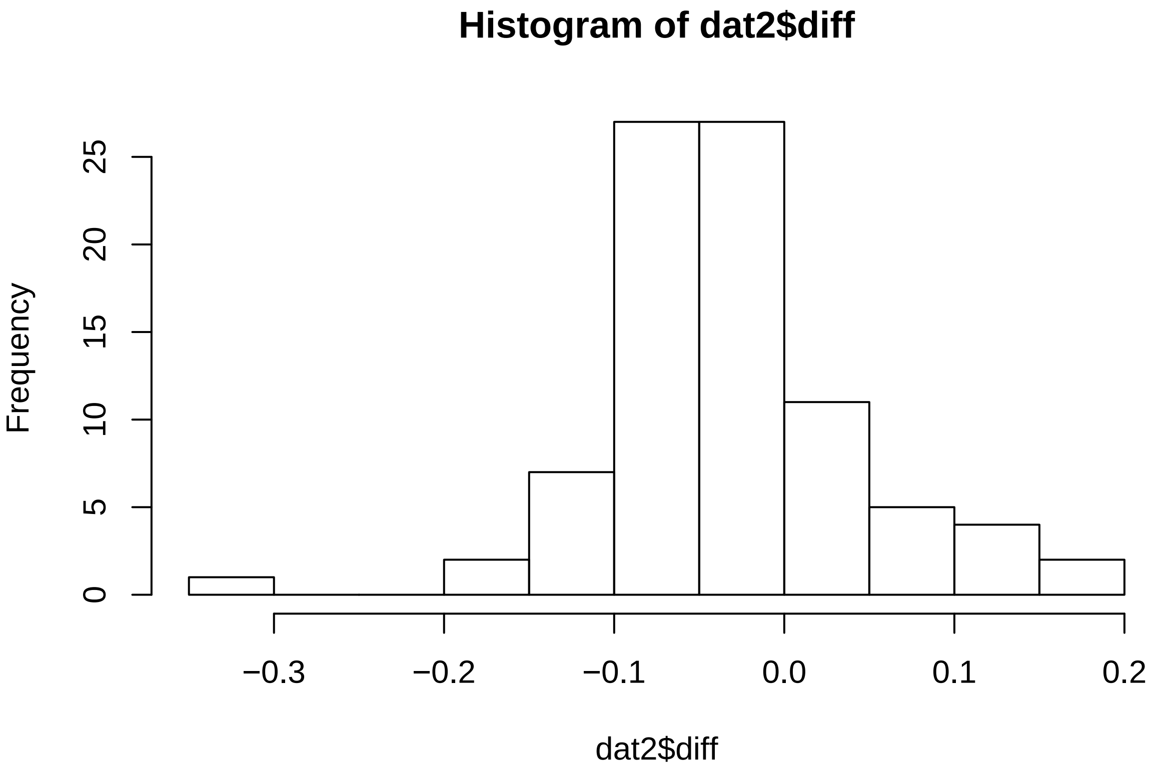
```
ggplot(data = dat2,
                    aes(x = canopygrowth, y = canopygrowth_f, colour= region))+
  geom_point()+
   geom_smooth(method = "lm")+
  labs(y=expression(paste(’Field-based canopy height growth (m year’^’-1’, ’)’)), x=expression(paste(’L
  theme_bw()+

  labs(colour="Region")+
  theme(text = element_text(size = 12))+
  ylim(0, 0.4)+
  theme(legend.position = ’right’,
                           legend.justification = c("left", "top"),
                           legend.box.just = "left",
                           #legend.margin = margin(5, 5, 5, 5),
                           legend.text = element_text(size=12))+
   geom_abline(intercept = 0, slope = 1, color="red",
                 linetype="dashed", size=1.5)
```



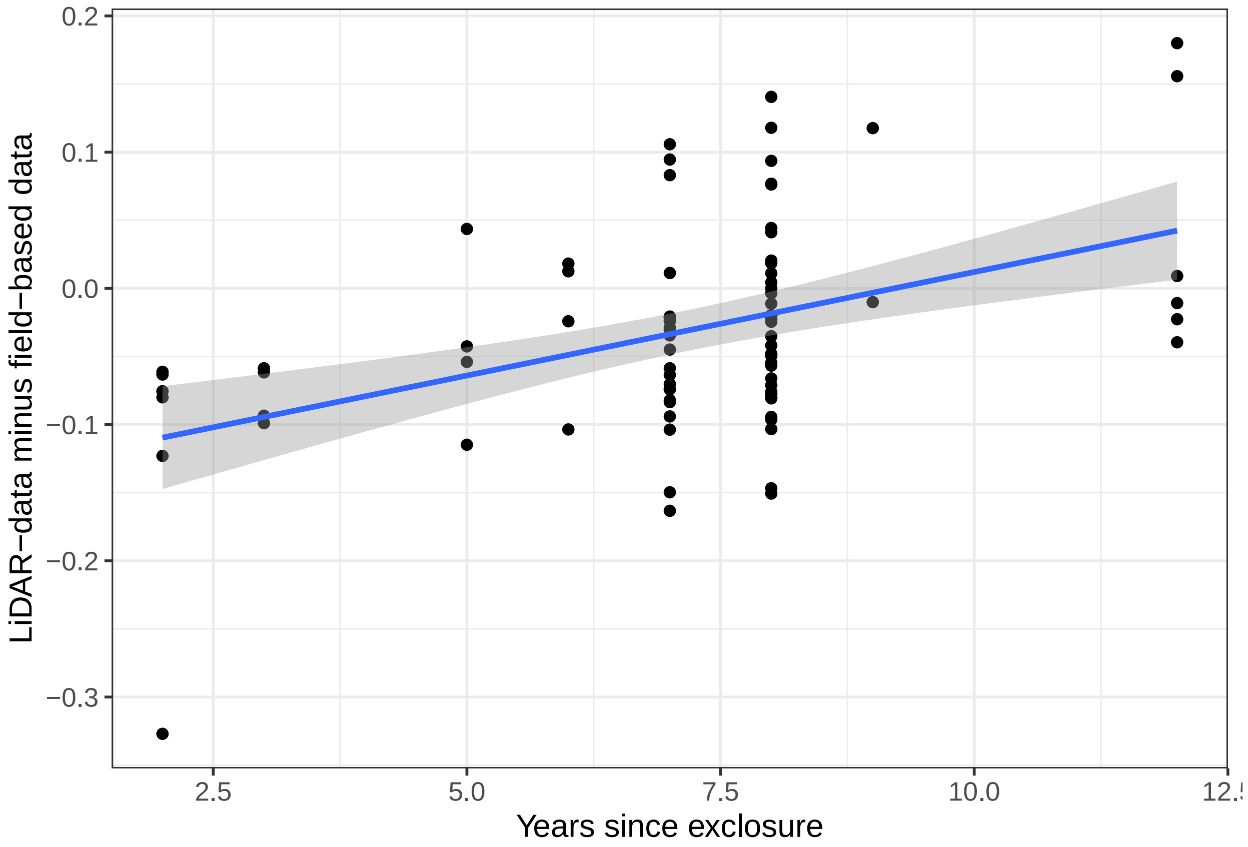
Also it differs a bit between region.

Also, we can look at the effect that experimental duration has had on the correlation:

```
dat2$diff <- dat2$canopygrowth-dat2$canopygrowth_f
hist(dat2$diff)
```



```
ggplot(data = dat2,
        aes(x = YrsSinceExclosure, y = diff))+
  geom_point()+
  geom_smooth(method = "lm")+
  labs(y="LiDAR-data minus field-based data",
         x="Years since exclosure")+
  theme_bw()+
  theme(text = element_text(size = 12))+
  #ylim(0, 0.4)+
  theme(legend.position = 'right',
                             legend.justification = c("left", "top"),
                             legend.box.just = "left",
                             #legend.margin = margin(5, 5, 5, 5),
                             legend.text = element_text(size=12))+
   geom_abline(intercept = 0, slope = 1, color="red",
                 linetype="dashed", size=1.5)
```



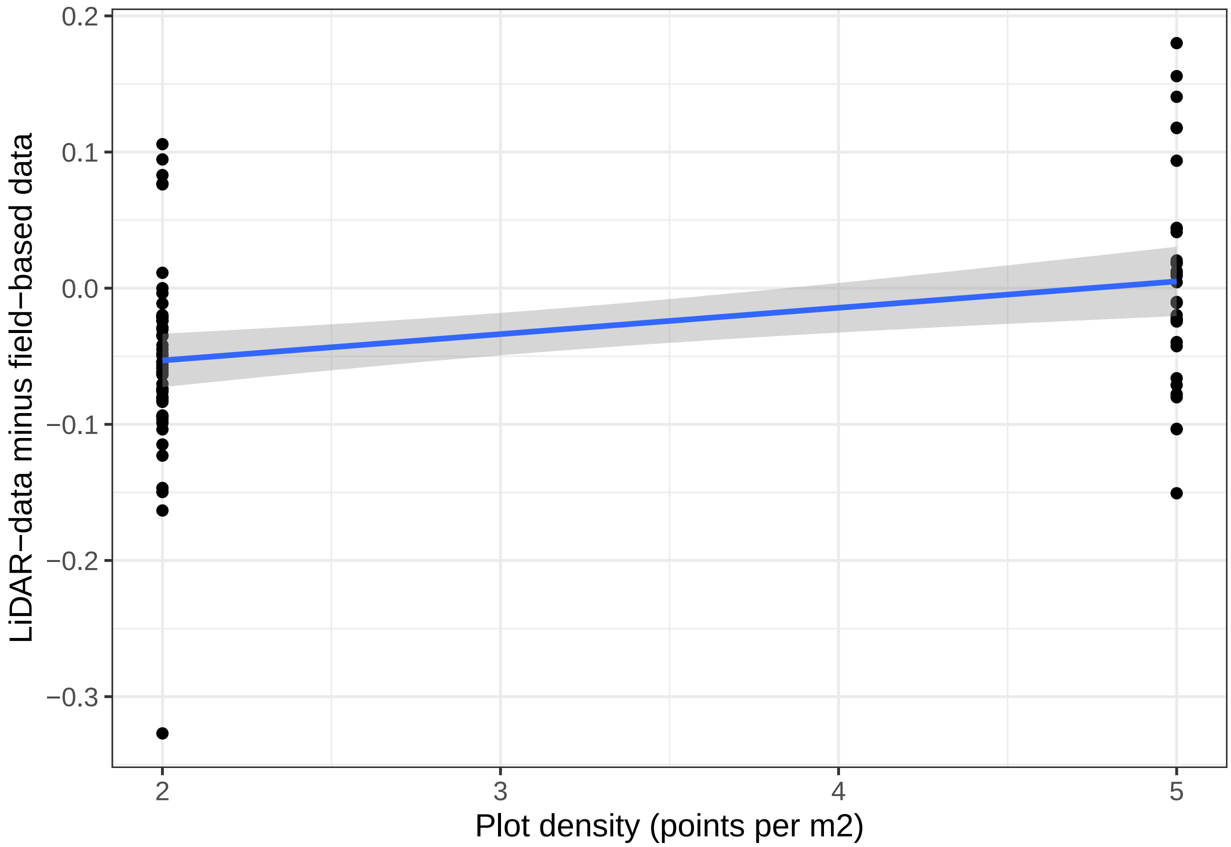
It looks like older exclosures have a greater difference between LiDAR and field-based data.

I would like to see if the correlation depends on LiDAR resolution, but then we should figure out what this is about:

```
table(dat2$plot_density_m2, dat2$resolution_m)
```

```
##
##     0,25 0,5
##   2    2  52
##   5   30   2
```

It's not clear which is derived from which, but I think its plot density that is the original.



No big difference.

Note that we are not seeing if the slope is close to 1 or not. We assume it will be. It's correlation strength that matters.

```
cor(dat2$canopygrowth, dat2$canopygrowth_f)
```

```
## [1] 0.5693676
```

And it's not very high. We can use a more sophisticated model to see where the field-based data is better or worse at predicting LiDAR results.

```
moddiff <- glmmTMB(canopygrowth~canopygrowth_f
                   +region+canopygrowth_f:region
                   +YrsSinceExclosure+YrsSinceExclosure:canopygrowth_f
                   +plot_density_m2+plot_density_m2:canopygrowth_f
                   +Treatment+Treatment:canopygrowth_f,
                   family=gaussian,
                   data=dat2)
summary(moddiff)
```

```
##  Family: gaussian  ( identity )
## Formula:
## canopygrowth ~ canopygrowth_f + region + canopygrowth_f:region +
##     YrsSinceExclosure + YrsSinceExclosure:canopygrowth_f + plot_density_m2 +
##     plot_density_m2:canopygrowth_f + Treatment + Treatment:canopygrowth_f
## Data: dat2
##
##      AIC      BIC   logLik deviance df.resid
##   -216.8   -184.9    121.4   -242.8       73
##
##
## Dispersion estimate for gaussian family (sigma^2): 0.00348
##
## Conditional model:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                        0.028696   0.067070   0.428    0.669
## canopygrowth_f                    -0.501168   0.437962  -1.144    0.252
## regionTelemark                    -0.021372   0.050695  -0.422    0.673
## regionTrondelag                   -0.027782   0.052147  -0.533    0.594
## YrsSinceExclosure                 -0.004944   0.010008  -0.494    0.621
## plot_density_m2                    0.004487   0.014719   0.305    0.760
## TreatmentExclosure                 0.021574   0.029502   0.731    0.465
## canopygrowth_f:regionTelemark      0.166680   0.400241   0.416    0.677
## canopygrowth_f:regionTrondelag     0.245282   0.404599   0.606    0.544
## canopygrowth_f:YrsSinceExclosure   0.109693   0.076821   1.428    0.153
## canopygrowth_f:plot_density_m2     0.077693   0.121632   0.639    0.523
## canopygrowth_f:TreatmentExclosure  0.141096   0.233017   0.606    0.545
```

Nothing significant. Applying dredge:

```r
library(MuMIn)
dredge(moddiff, beta="none", rank = "AICc")[1:5,]
```

```
## Fixed terms are "cond((Int))" and "disp((Int))"
```

```
## Global model call: glmmTMB(formula = canopygrowth ~ canopygrowth_f + region + canopygrowth_f:region +
##     YrsSinceExclosure + YrsSinceExclosure:canopygrowth_f + plot_density_m2 +
##     plot_density_m2:canopygrowth_f + Treatment + Treatment:canopygrowth_f,
##     data = dat2, family = gaussian, ziformula = ~0, dispformula = ~1)
## ---
## Model selection table
##     cnd((Int)) dsp((Int)) cnd(cnp_f) cnd(plt_dns_m2) cnd(Trt)  cnd(YSE)
## 284   0.002381          +    -0.2593        0.013230        + -0.008476
## 412   0.012060          +    -0.3450        0.013200        + -0.009152
## 316   0.006928          +    -0.2972        0.009724        + -0.007803
## 282   0.014310          +    -0.2343                        + -0.004758
## 444   0.015780          +    -0.3757        0.009960        + -0.008496
##     cnd(cnp_f:plt_dns_m2) cnd(cnp_f:Trt) cnd(cnp_f:YSE) df  logLik   AICc delta
## 284                                                0.1402  7 121.087 -226.7  0.00
## 412                                     +          0.1453  8 121.176 -224.5  2.26
## 316               0.02881                          0.1351  8 121.147 -224.4  2.31
## 282                                                0.1408  6 117.585 -222.1  4.63
## 444               0.02662               +          0.1403  9 121.227 -222.1  4.65
##     weight
## 284  0.545
## 412  0.176
## 316  0.171
## 282  0.054
## 444  0.053
## Models ranked by AICc(x)
```

And the best model is…:

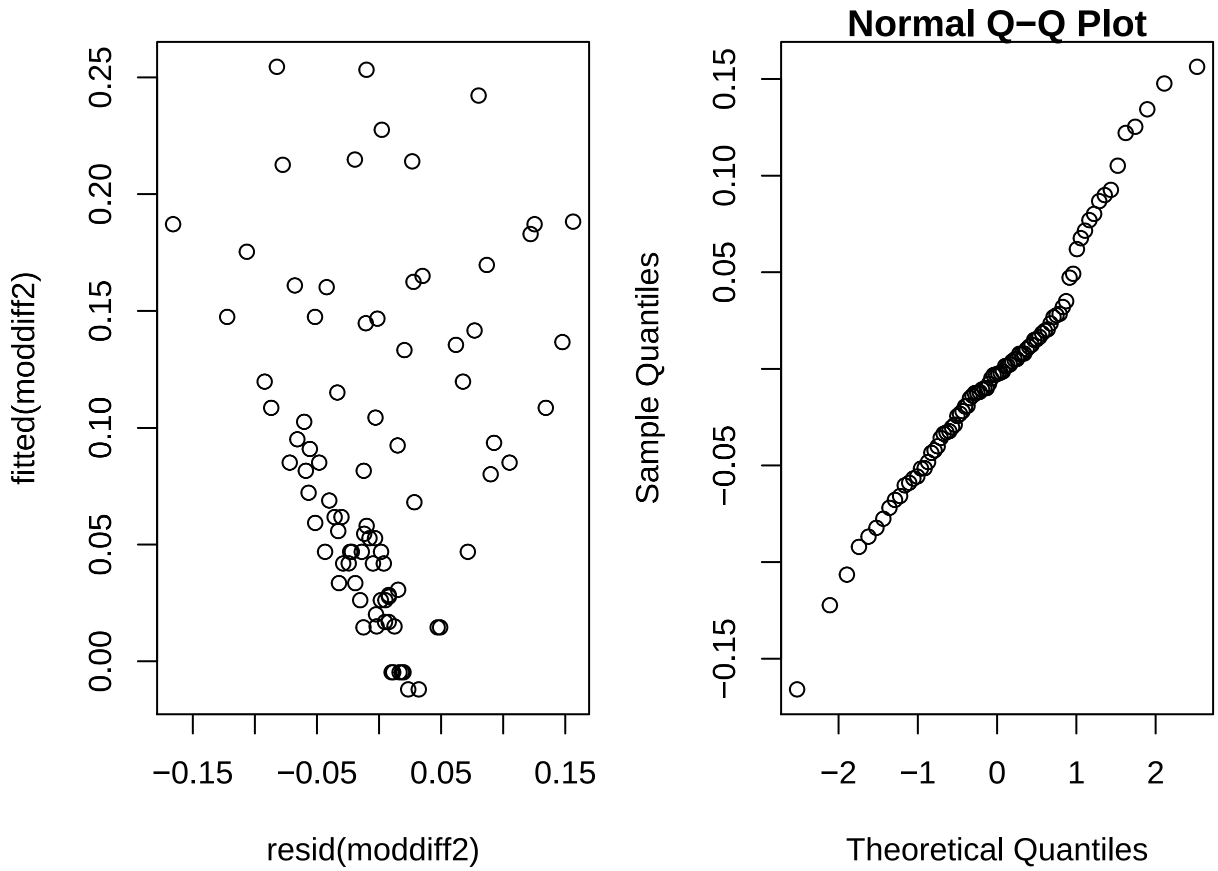
```
moddiff2 <- glmmTMB(canopygrowth~canopygrowth_f
                   +YrsSinceExclosure+YrsSinceExclosure:canopygrowth_f
                   +plot_density_m2
                   +Treatment,
                   family=gaussian,
                   data=dat2)
summary(moddiff2)
```

```
##  Family: gaussian  ( identity )
## Formula:
## canopygrowth ~ canopygrowth_f + YrsSinceExclosure + YrsSinceExclosure:canopygrowth_f +
##     plot_density_m2 + Treatment
## Data: dat2
##
##      AIC      BIC   logLik deviance df.resid
##   -228.2   -211.0    121.1   -242.2       79
##
##
## Dispersion estimate for gaussian family (sigma^2): 0.0035
##
## Conditional model:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      0.002381   0.047428   0.050  0.95997
## canopygrowth_f                  -0.259305   0.303197  -0.855  0.39242
## YrsSinceExclosure               -0.008476   0.006628  -1.279  0.20100
## plot_density_m2                  0.013229   0.004898   2.701  0.00691 **
## TreatmentExclosure               0.038179   0.013724   2.782  0.00540 **
## canopygrowth_f:YrsSinceExclosure  0.140224   0.043695   3.209  0.00133 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

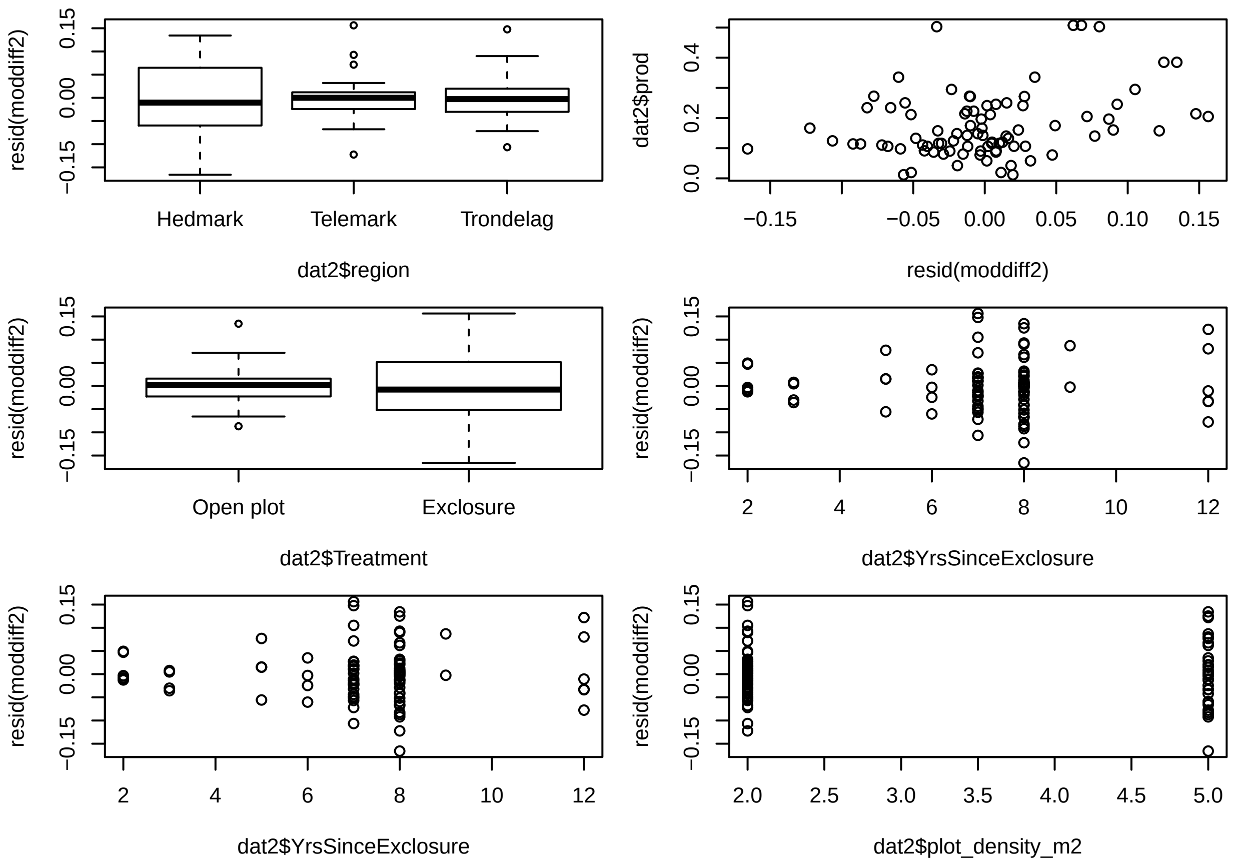
Interpretation: Field-based data have a medium correlation to LiDAR data (0.57). The coorelation is dependent on Years since exclosure (probably this could be exchanged with tree height?) and LiDAR data is more likely to underestimate canopy growth at short experimental durations (see figure above). Similarly, lower LiDAR resolution induces underestimates in the same way. Field-based data is also not able to explain the variation induced by the fencing treatemtn. This could be a three way ineraction as well, including years since exclusion. Not sure if it should be included here actually.

Validation

```
par(mfrow=c(1,2), mar = c(4,4,1,1))
plot(resid(moddiff2), fitted(moddiff2))
qqnorm(resid(moddiff2))
```

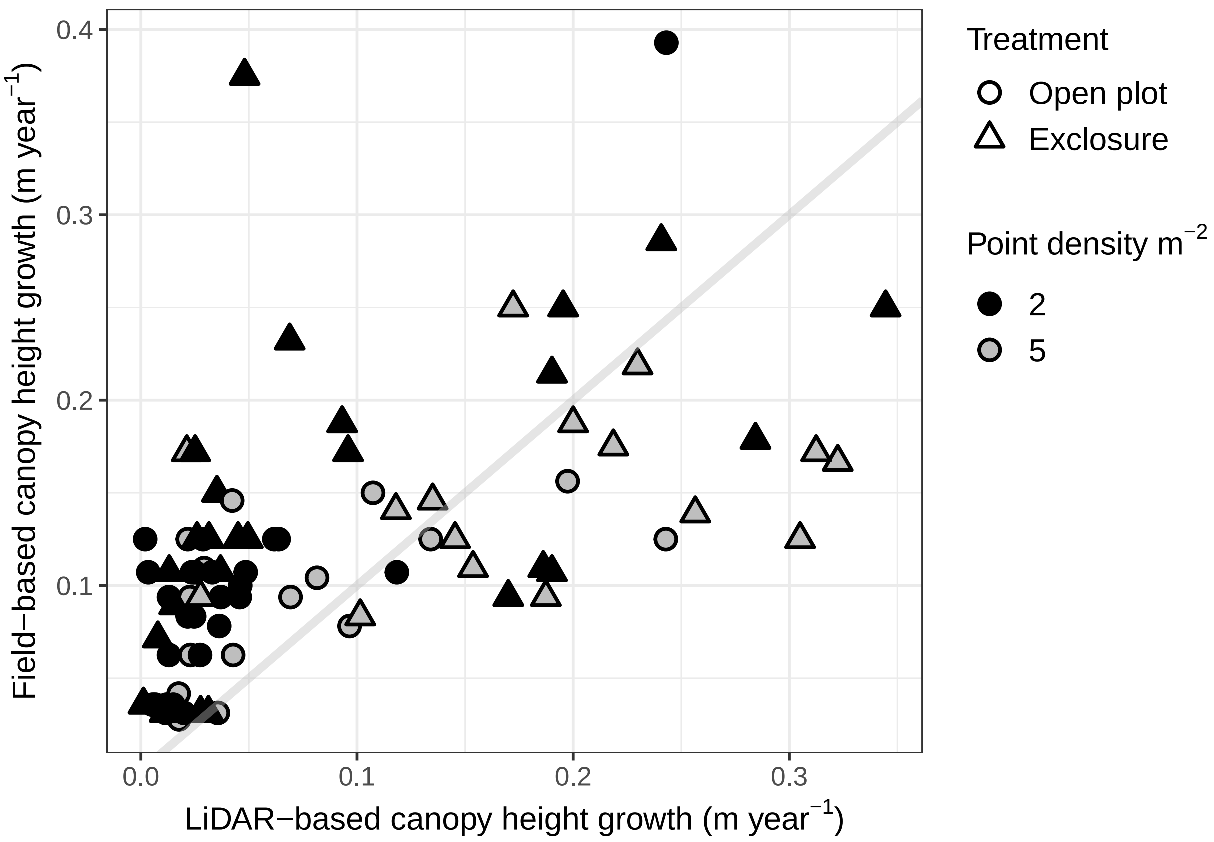


```r
par(mfrow=c(3,2), mar = c(4,4,1,1))
plot(resid(moddiff2)~ dat2$region)
plot(resid(moddiff2), dat2$prod)
plot(resid(moddiff2)~ dat2$Treatment)
plot(resid(moddiff2)~ dat2$YrsSinceExclosure)
plot(resid(moddiff2)~ dat2$YrsSinceExclosure)
plot(resid(moddiff2)~ dat2$plot_density_m2)
```



Looks Ok For the final plot I think just keep it simple.

```
ggplot(data = dat2,
                    aes(x = canopygrowth, y = canopygrowth_f))+
   geom_point(aes(shape = Treatment,
                  fill = as.factor(plot_density_m2)),
                  size=3, stroke=1)+
  labs(y=expression(paste(’Field-based canopy height growth (m year’^’-1’, ’)’)),
       x=expression(paste(’LiDAR-based canopy height growth (m year’^’-1’, ’)’)))+
  theme_bw()+
  theme(text = element_text(size = 12))+
  theme(legend.position = ’right’,
                             legend.justification = c("left", "top"),
                             legend.box.just = "left",
                             #legend.margin = margin(5, 5, 5, 5),
                             legend.text = element_text(size=12))+
  scale_shape_manual(values=c(21, 24), name = "Treatment")+
  scale_fill_manual(values = c("black", "grey"),
                    name = expression(paste(ext="Point density m"^"-2")))+
  geom_abline(intercept = 0, slope = 1, color="grey",
                 linetype="solid", size=1.5, alpha  = 0.4)+
  guides(fill = guide_legend(override.aes=list(shape=21)))
```



Thta last line is tricky (https://github.com/tidyverse/ggplot2/issues/2322)

Treatment differences