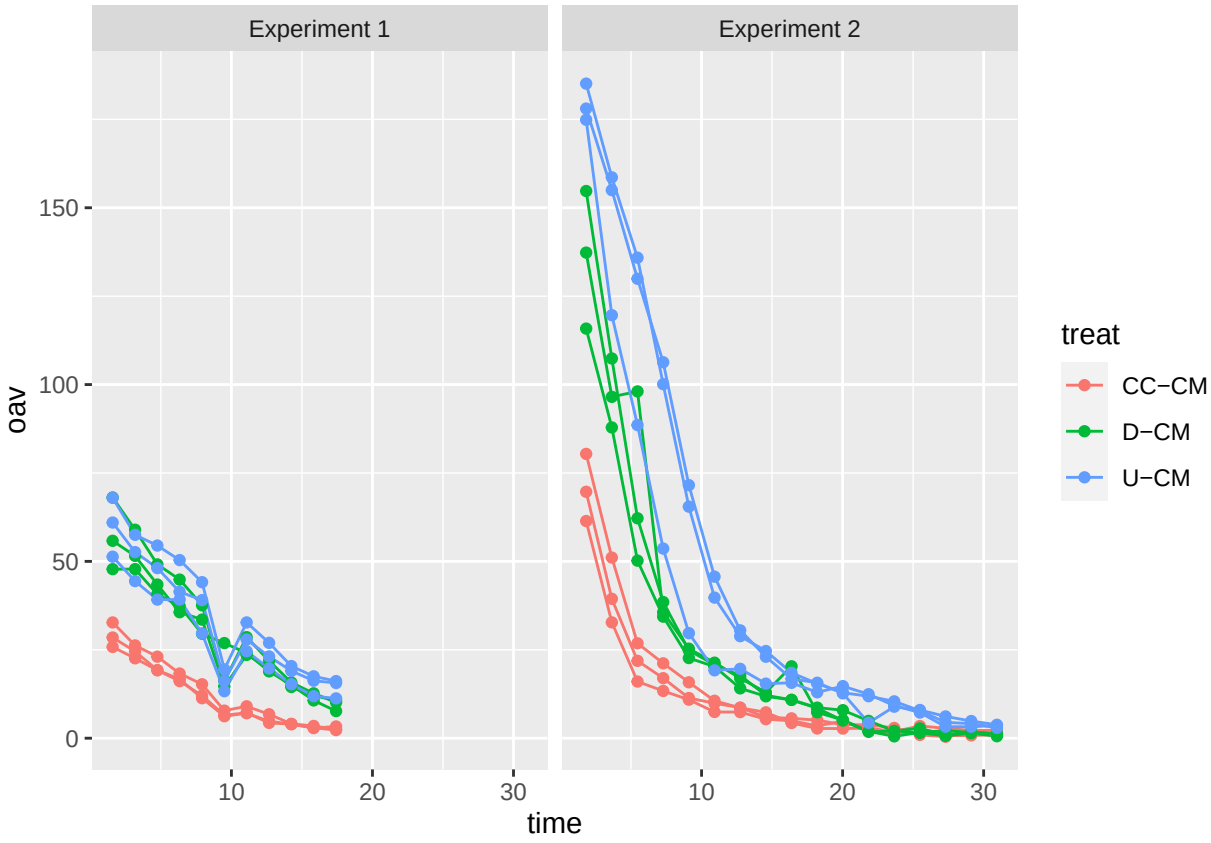
Data analysis for odor from digestate experiments

Sasha D. Hafner

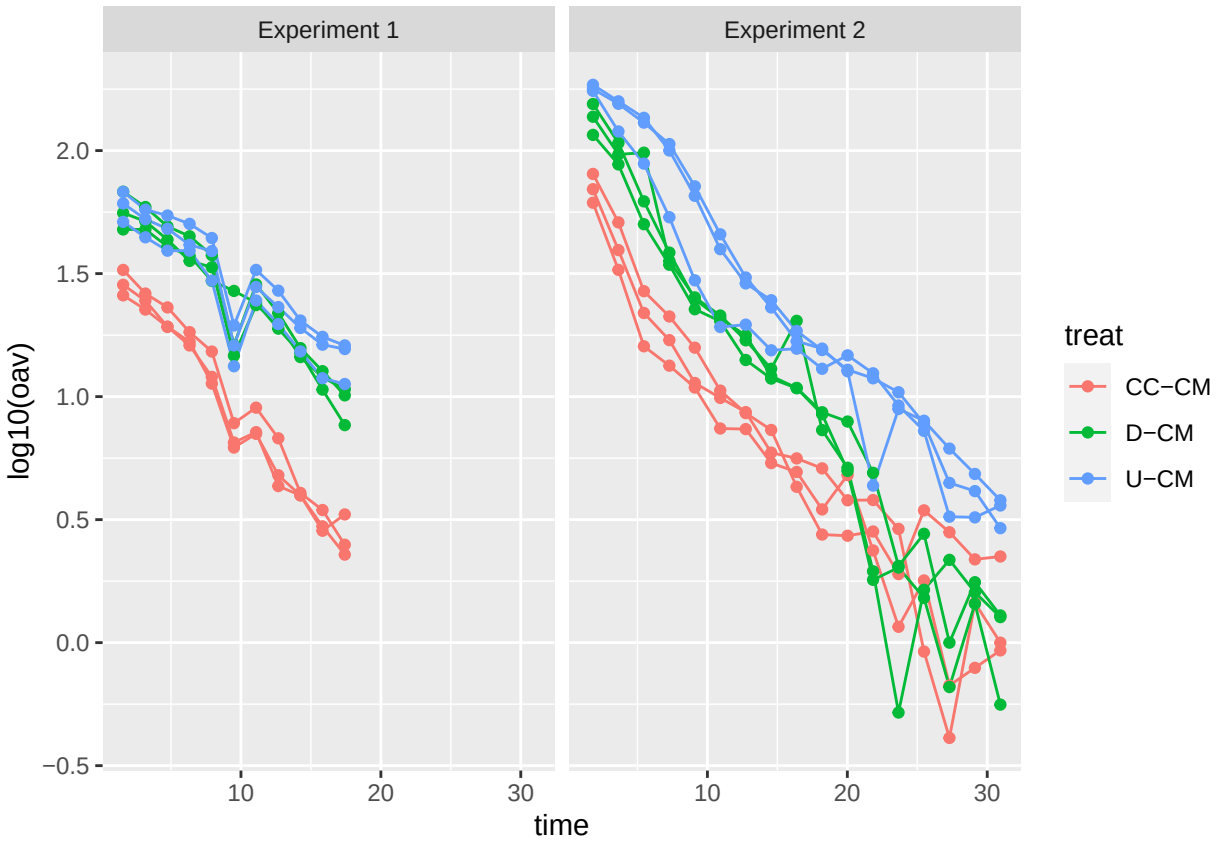
01 December, 2022

Plots

```
ggplot(dat, aes(time, oav, colour = treat, group = interaction(rep, treat))) +
  geom_line() +
  geom_point() +
  facet_wrap(~ exper)
```



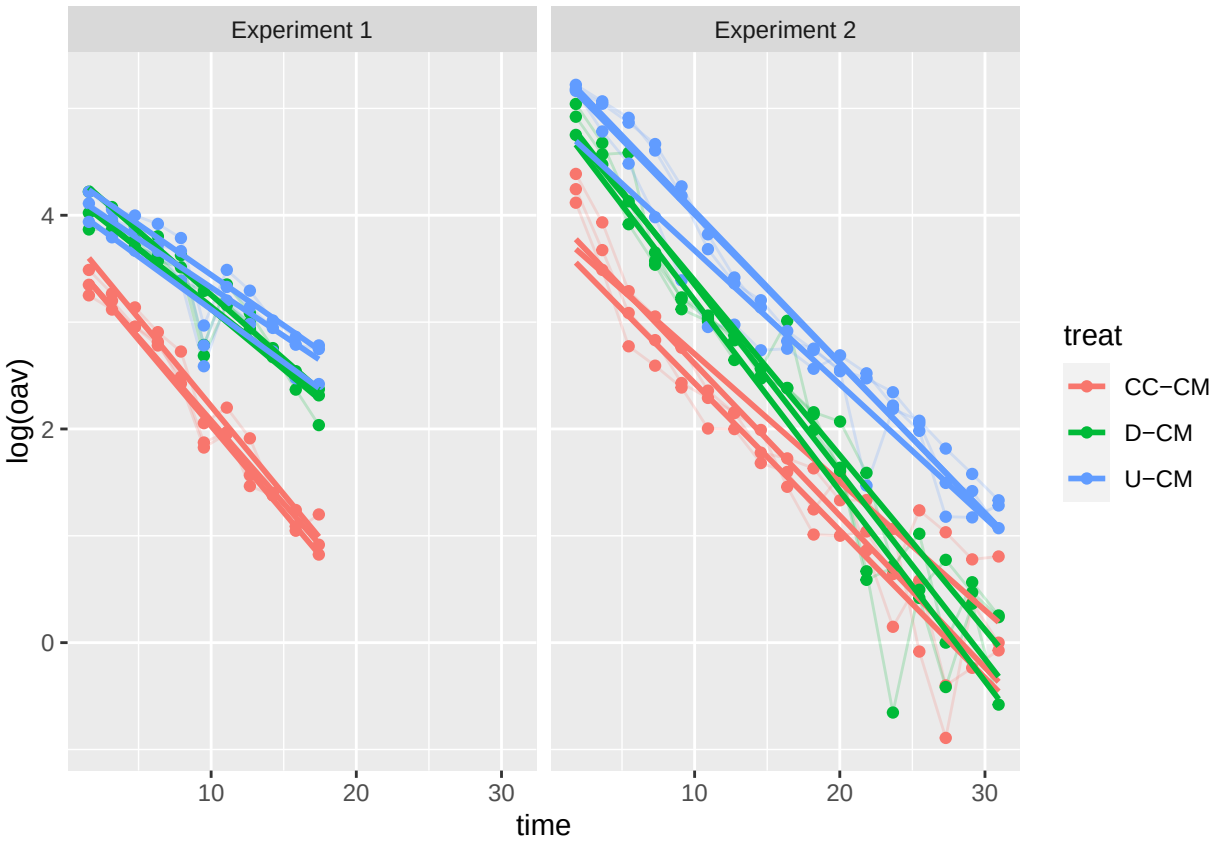

```
ggplot(dat, aes(time, log10(oav), colour = treat, group = interaction(rep, treat))) +
  geom_line() +
  geom_point() +
  facet_wrap(~ exper)
```



Looks linear enough.

```
ggplot(dat, aes(time, log(oav), colour = treat, group = interaction(rep, treat))) +
  geom_line(alpha = 0.2) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE) +
  facet_wrap(~ exper)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Stats

So, our question will be (referring to last plot above) "are there differences in slope and initial value?". I had expected to use the intercept and slope terms in the analysis, but the least-squares lines don't fit very well in all cases at the start. And anyway, the initial measurements are our best estimate of initial OAV.

Set reference to untreated cattle manure.

```
dat$treat <- factor(dat$treat, levels = c('U-CM', 'CC-CM', 'D-CM'))
dat$exper <- factor(dat$exper)
```

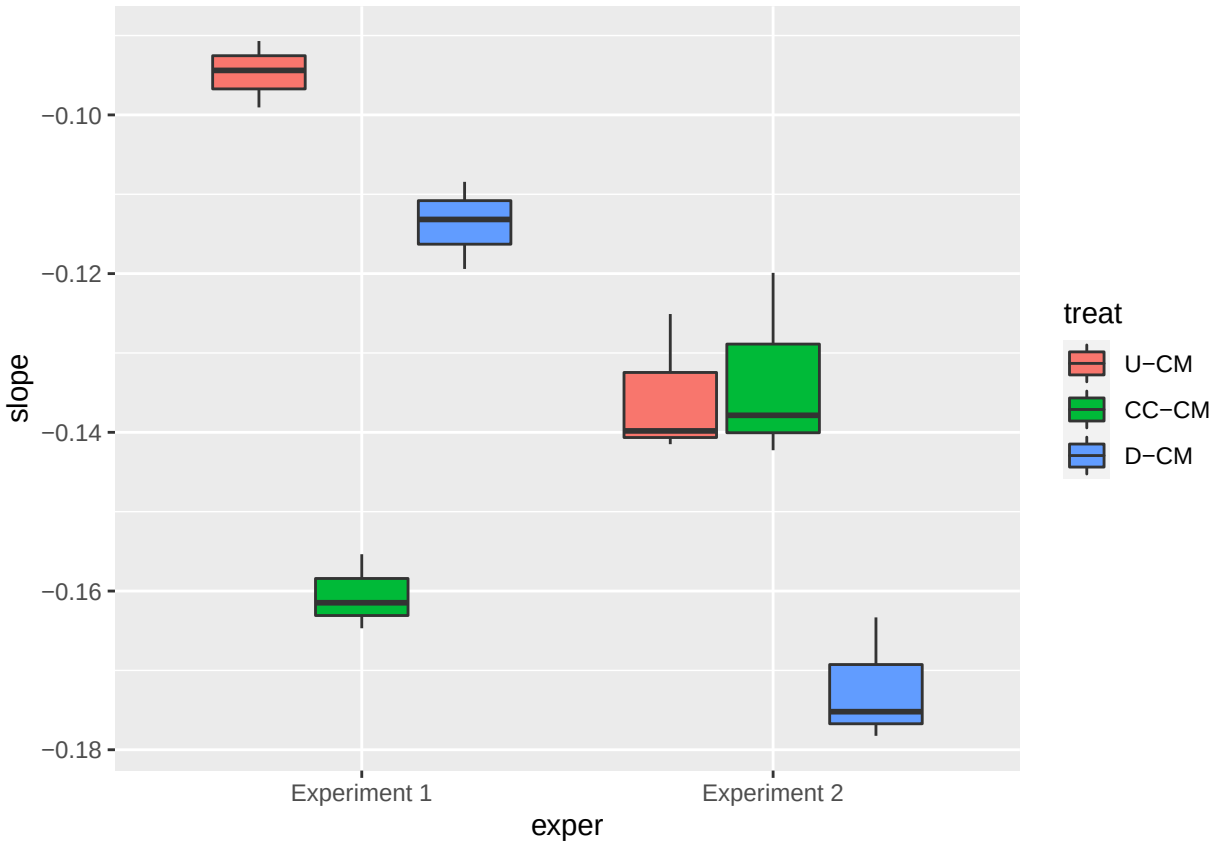
Unit of analysis will be wind tunnel plot.

First fit linear model to each wind tunnel to get the slopes.

```
lmods <- dat[, .(int = coef(lm(log(oav) ~ time))[1],
                 slope = coef(lm(log(oav) ~ time))[2]),
             by = .(exper, treat, rep)]
```

Take a look at slopes.

```
ggplot(lmods, aes(exper, slope, fill = treat)) +
  geom_boxplot()
```



Clear differences for D-CM in both experiments, but no difference for CC-CM in experiment 2.

Now analysis.

Look at slope.

```
modslope1 <- lm(slope ~ treat * exper, data = lmods)
summary.aov(modslope1)
```

```
##             Df   Sum Sq  Mean Sq F value   Pr(>F)
## treat        2 0.003612 0.001806   30.60 1.94e-05 ***
## exper        1 0.002604 0.002604   44.12 2.39e-05 ***
## treat:exper  2 0.006143 0.003072   52.04 1.22e-06 ***
## Residuals   12 0.000708 0.000059
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(modslope1)
```

```
##
## Call:
## lm(formula = slope ~ treat * exper, data = lmods)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.008916 -0.004477 -0.001957  0.004868  0.013433
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)                  -0.094712   0.004436 -21.352 6.49e-11 ***
## treatCC-CM                   -0.065799   0.006273 -10.489 2.13e-07 ***
## treatD-CM                    -0.018960   0.006273  -3.022   0.0106 *
## experExperiment 2            -0.040756   0.006273  -6.497 2.95e-05 ***
## treatCC-CM:experExperiment 2  0.067930   0.008872   7.657 5.87e-06 ***
## treatD-CM:experExperiment 2  -0.017829   0.008872  -2.010   0.0675 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.007683 on 12 degrees of freedom
## Multiple R-squared:  0.9458, Adjusted R-squared:  0.9232
## F-statistic: 41.88 on 5 and 12 DF,  p-value: 3.463e-07
```

Interactions complicated. Let's look by experiment. First experiment 1.

```
modexp1 <- aov(slope ~ treat, data = lmods, subset = exper == 'Experiment 1')
summary(modexp1)
```

```
##             Df   Sum Sq  Mean Sq F value   Pr(>F)
## treat        2 0.006883 0.003441   146.8 8.04e-06 ***
## Residuals    6 0.000141 0.000023
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary.lm(modexp1)
```

```
##
## Call:
## aov(formula = slope ~ treat, data = lmods, subset = exper ==
##     "Experiment 1")
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0057368 -0.0041828  0.0003195  0.0040237  0.0052372
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.094712   0.002796 -33.877  4.4e-08 ***
## treatCC-CM  -0.065799   0.003954 -16.642  3.0e-06 ***
## treatD-CM   -0.018960   0.003954  -4.796  0.00301 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.004842 on 6 degrees of freedom
## Multiple R-squared:   0.98,  Adjusted R-squared:  0.9733
## F-statistic: 146.8 on 2 and 6 DF,  p-value: 8.037e-06
```

```
coef(modexp1)
```

```
## (Intercept)  treatCC-CM   treatD-CM
## -0.09471158 -0.06579903 -0.01896045
```

```
confint(modexp1)
```

```
##                    2.5 %       97.5 %
## (Intercept) -0.10155248 -0.087870677
## treatCC-CM  -0.07547353 -0.056124528
```

```
## treatD-CM   -0.02863495 -0.009285954
```

```
model.tables(modexp1, type = 'means')
```

```
## Tables of means
## Grand mean
##
## -0.1229647
##
##  treat
## treat
##     U-CM    CC-CM     D-CM
## -0.09471 -0.16051 -0.11367
```

Use this model in paper. Both D-CM and CC-CM have lower slope than reference in experiment 1. Results are a first-order constant with units of 1/hr that describes the drop in flux over time. Note that `log()` (not `log10()`) used above gives this result.

Experiment 2 next.

```
modexp2 <- aov(slope ~ treat, data = lmods, subset = exper == 'Experiment 2')
summary(modexp2)
```

```
##             Df    Sum Sq   Mean Sq F value  Pr(>F)
## treat        2 0.0028727 0.0014364   15.18 0.00449 **
## Residuals    6 0.0005676 0.0000946
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary.lm(modexp2)
```

```
##
## Call:
## aov(formula = slope ~ treat, data = lmods, subset = exper ==
##     "Experiment 2")
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.008916 -0.005988 -0.004354  0.008936  0.013433
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.135468   0.005616 -24.123 3.33e-07 ***
## treatCC-CM   0.002131   0.007942   0.268  0.79745
## treatD-CM   -0.036789   0.007942  -4.632  0.00357 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.009727 on 6 degrees of freedom
## Multiple R-squared:  0.835,  Adjusted R-squared:   0.78
## F-statistic: 15.18 on 2 and 6 DF,  p-value: 0.004492
```

```
coef(modexp2)
```

```
##  (Intercept)   treatCC-CM    treatD-CM
## -0.135468016  0.002130819 -0.036788980
```

```r
confint(modexp2)
```

```
##                   2.5 %      97.5 %
## (Intercept) -0.14920914 -0.12172689
## treatCC-CM  -0.01730207  0.02156370
## treatD-CM   -0.05622187 -0.01735609
```

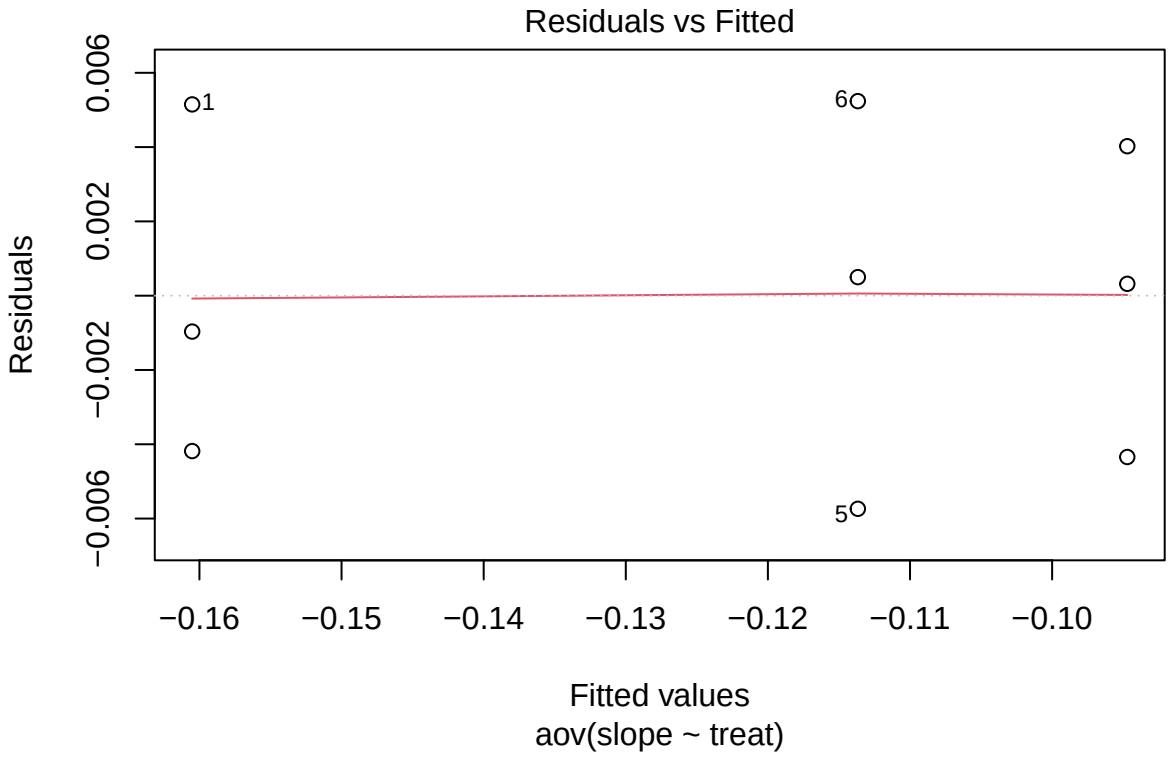
```r
model.tables(modexp2, type = 'means')
```

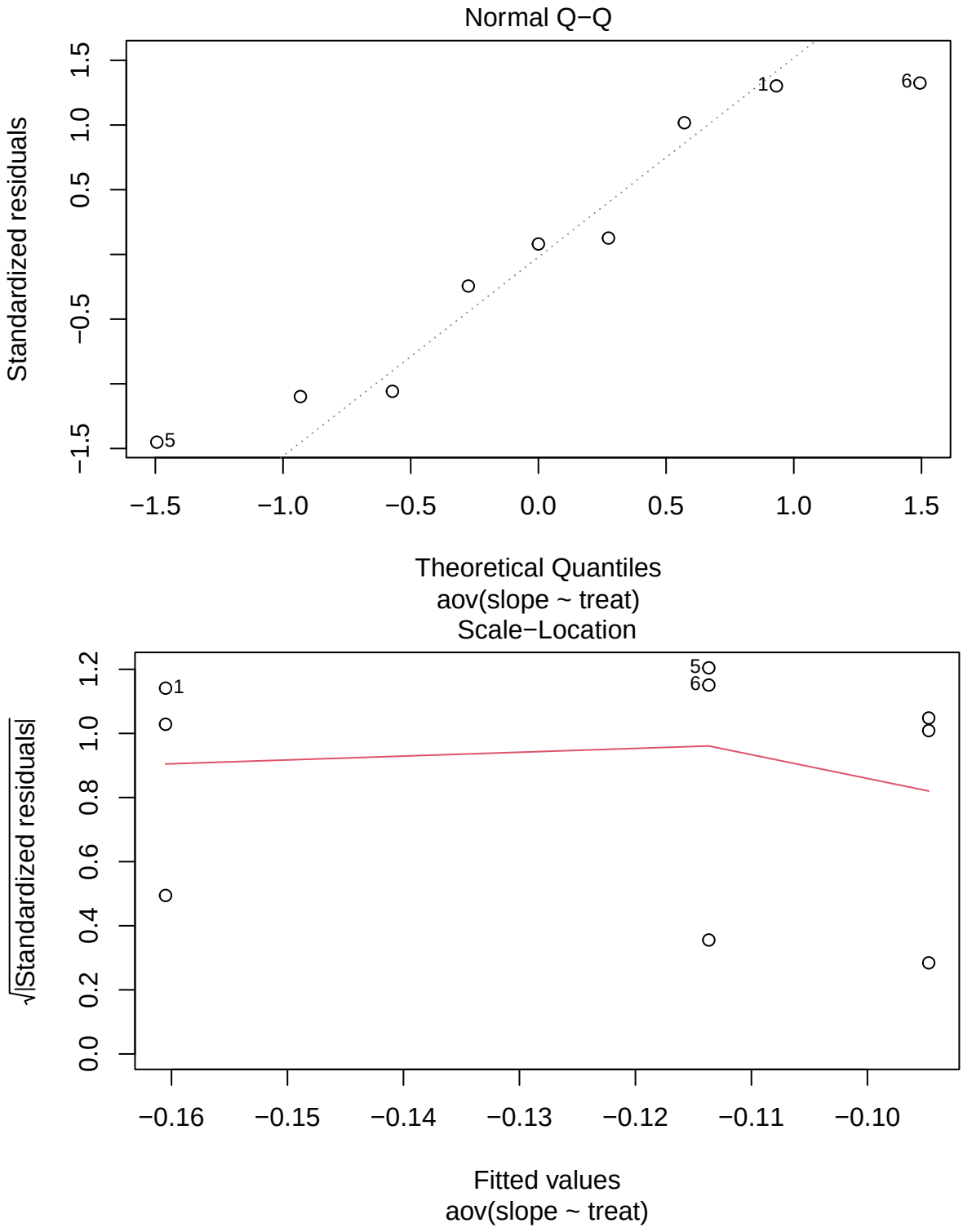
```
## Tables of means
## Grand mean
##
## -0.1470207
##
##  treat
## treat
##     U-CM    CC-CM     D-CM
## -0.13547 -0.13334 -0.17226
```

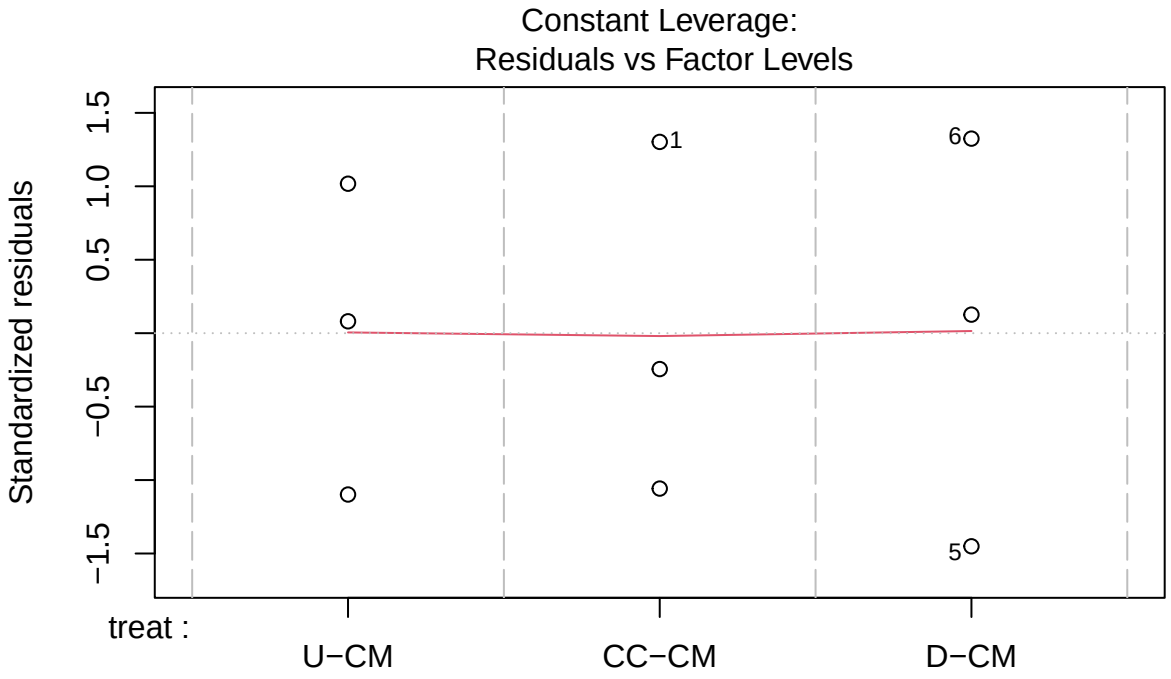
Here only D-CM is lower, and the effect seems larger than in experiment 1. **Use this model in paper for experiment 2.**

Diagnostic plots.

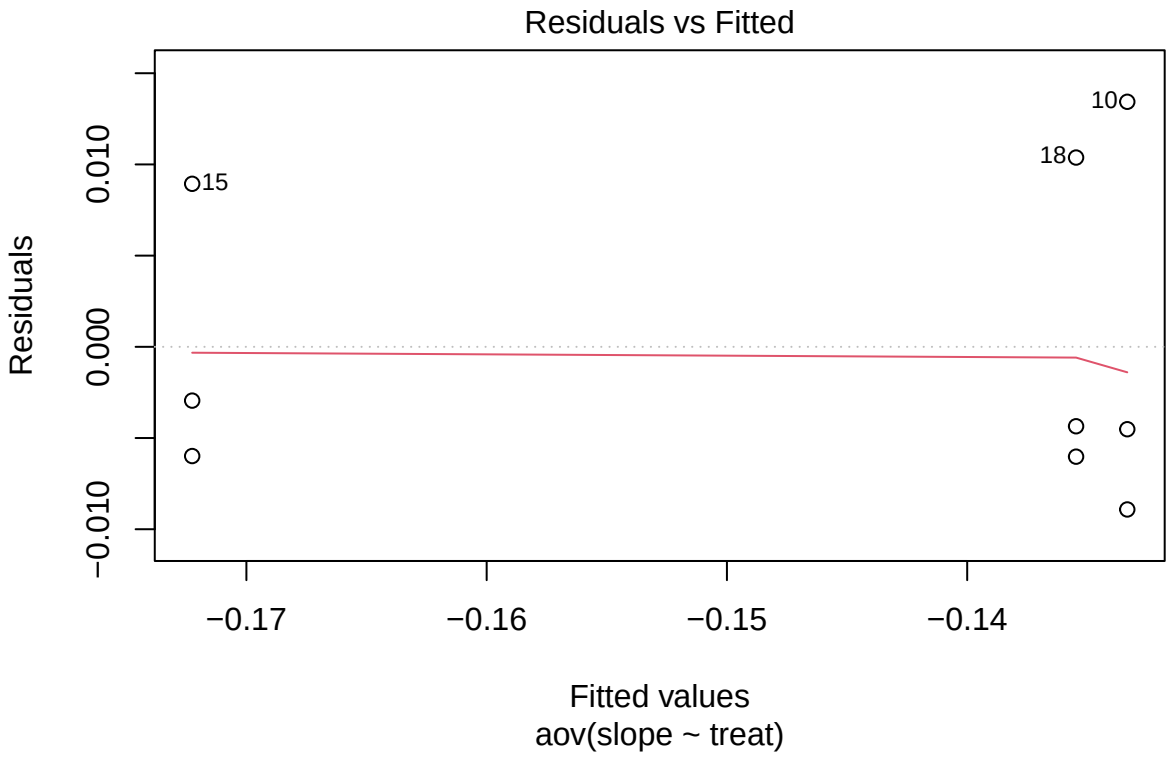
```r
plot(modexp1, ask = FALSE)
```

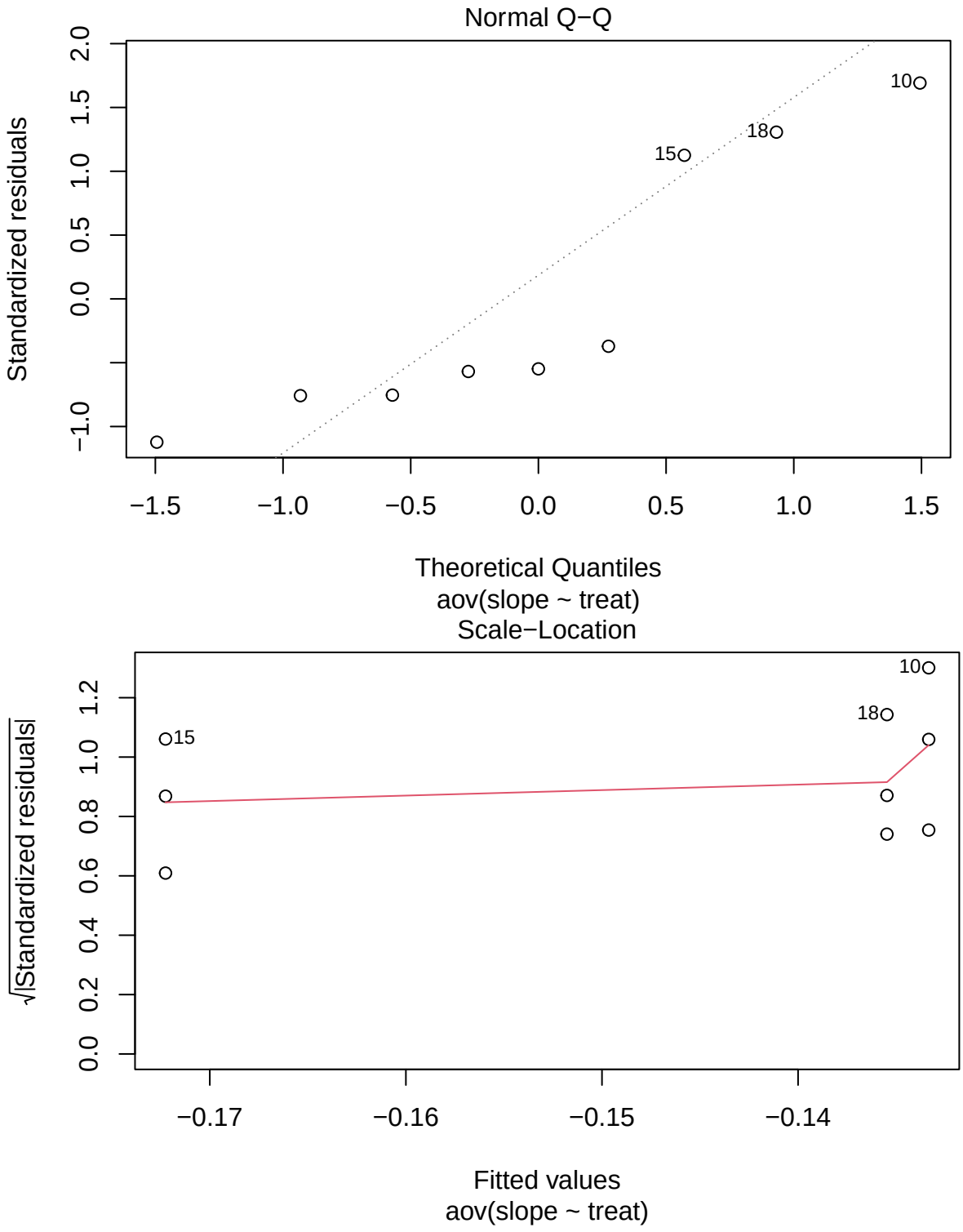


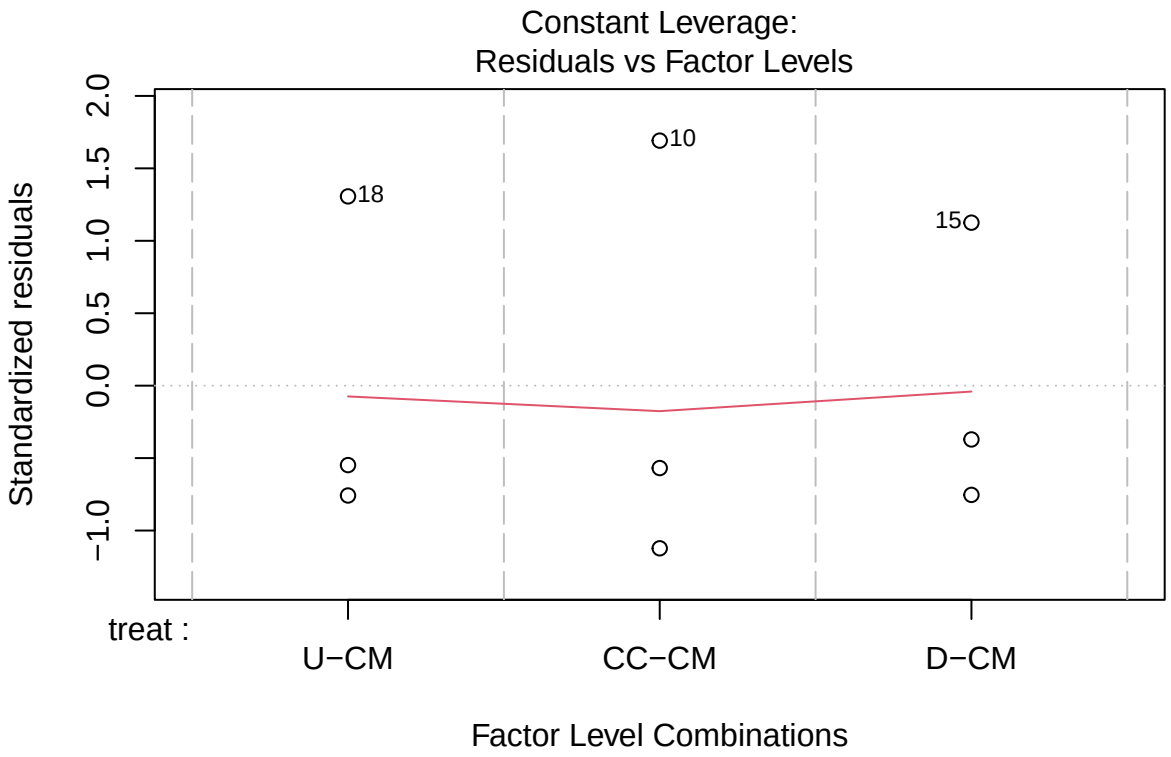

Normal Q−Q
Standardized residuals
Theoretical Quantiles
aov(slope ~ treat)
Scale−Location
√|Standardized residuals|
Fitted values
aov(slope ~ treat)




```
plot(modexp2, ask = FALSE)
```




Normal Q−Q
Standardized residuals
2.0
1.5
1.0
0.5
0.0
−1.0
10
18
15
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Theoretical Quantiles
aov(slope ~ treat)
Scale−Location
√|Standardized residuals|
1.2
1.0
0.8
0.6
0.4
0.2
0.0
10
18
15
−0.17
−0.16
−0.15
−0.14
Fitted values
aov(slope ~ treat)



Not terrible.

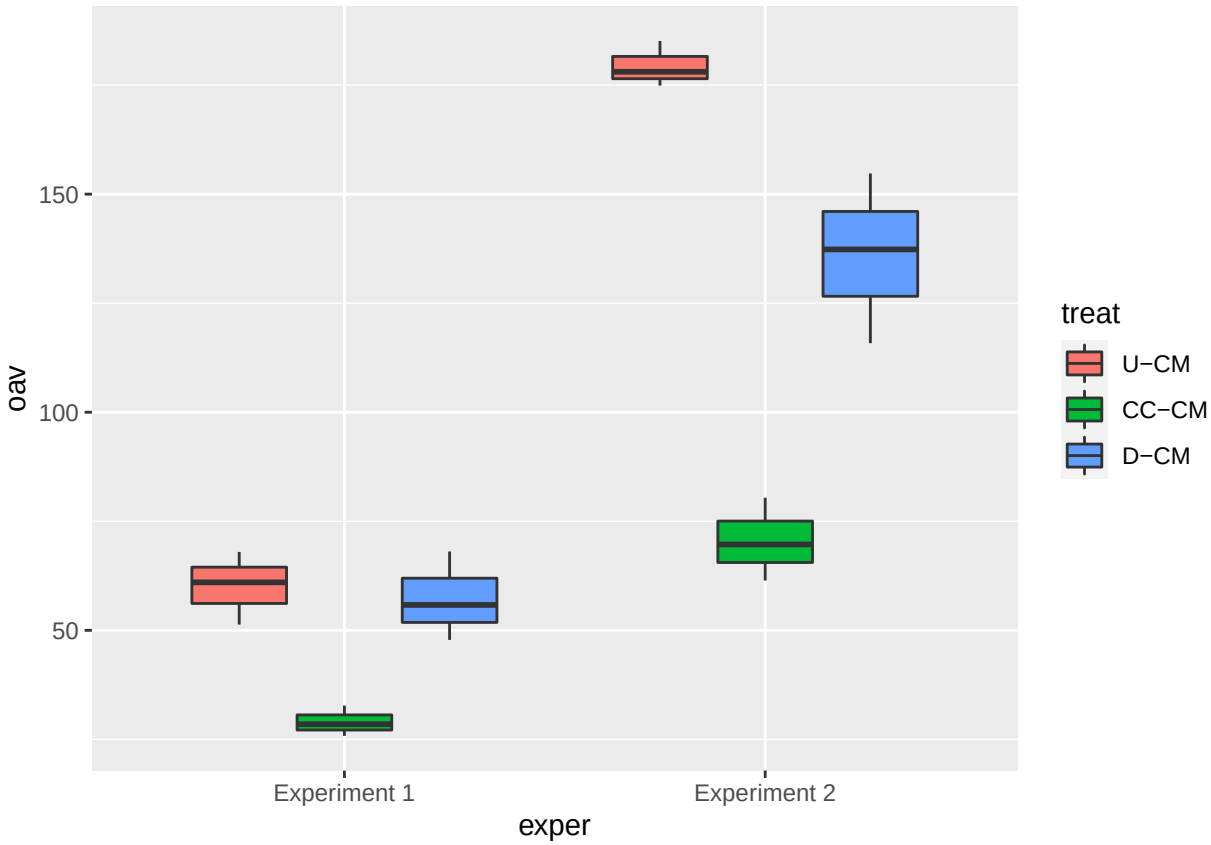
Now look at first measurement period in lieu of intercept.

Add interval number to data.

```
dat <- dat[, int := as.integer(factor(time)), by = exper]
```

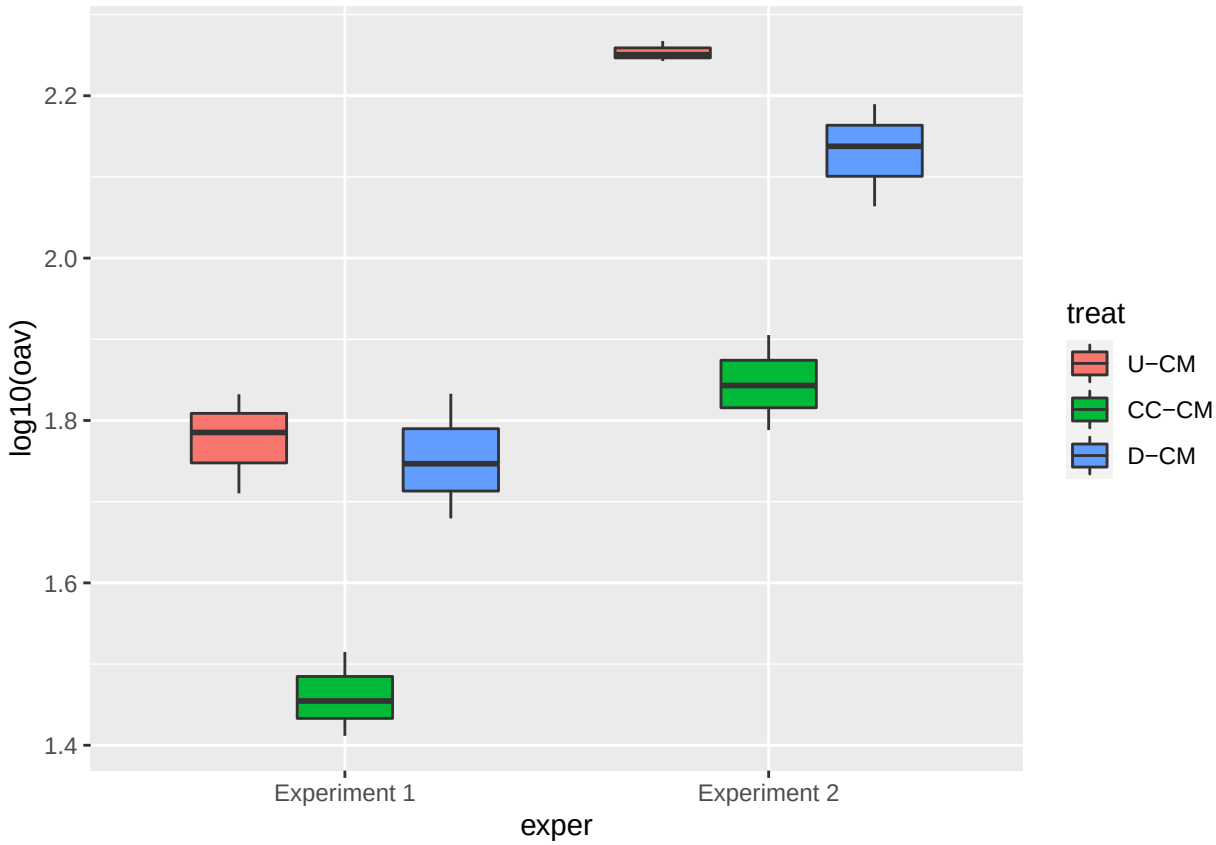
And take a look at the initial OAV values.

```
dd <- subset(dat, int == 1)
ggplot(dd, aes(exper, oav, fill = treat)) +
  geom_boxplot()
```



Check transformed values.

```
ggplot(dd, aes(exper, log10(oav), fill = treat)) +
  geom_boxplot()
```



See CC-CM lower in both experiments, D-CM lower in experiment 2 only.

```
modinit1 <- aov(log10(oav) ~ treat * exper, data = dat, subset = int == 1)
summary(modinit1)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## treat        2 0.4392  0.2196  65.935 3.37e-07 ***
## exper        1 0.7689  0.7689 230.841 3.36e-09 ***
## treat:exper  2 0.0093  0.0047   1.401    0.284
## Residuals   12 0.0400  0.0033
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Drop interaction.

```
modinit2 <- aov(log10(oav) ~ treat + exper, data = dat, subset = int == 1)
summary(modinit2)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## treat        2 0.4392  0.2196   62.37 1.07e-07 ***
## exper        1 0.7689  0.7689  218.34 6.21e-10 ***
## Residuals   14 0.0493  0.0035
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary.lm(modinit2)
```

```
##
## Call:
## aov(formula = log10(oav) ~ treat + exper, data = dat, subset = int ==
```

```
##     1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.09779 -0.03169  0.01001  0.03815  0.09792
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.80807    0.02797  64.633  < 2e-16 ***
## treatCC-CM        -0.36180    0.03426 -10.560 4.74e-08 ***
## treatD-CM         -0.07304    0.03426  -2.132   0.0512 .
## experExperiment 2  0.41336    0.02797  14.776 6.21e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05934 on 14 degrees of freedom
## Multiple R-squared:  0.9608, Adjusted R-squared:  0.9524
## F-statistic: 114.4 on 3 and 14 DF,  p-value: 4.398e-10
```

```
100 * (1 - 10^coef(modinit2))
```

```
##       (Intercept)        treatCC-CM         treatD-CM experExperiment 2
##       -6327.96660          56.52926          15.47938        -159.03693
```

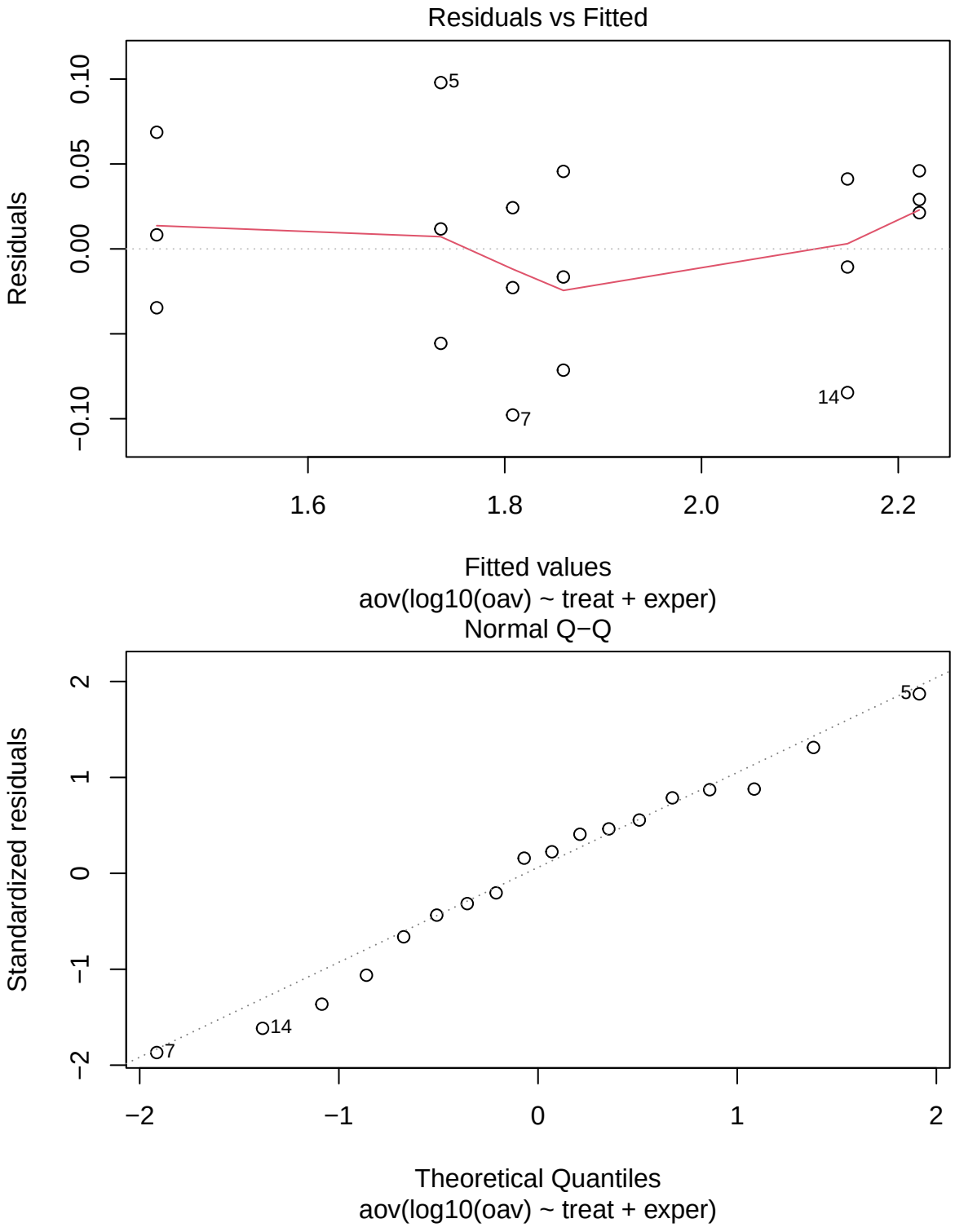
```
100 * (1 - 10^confint(modinit2))
```

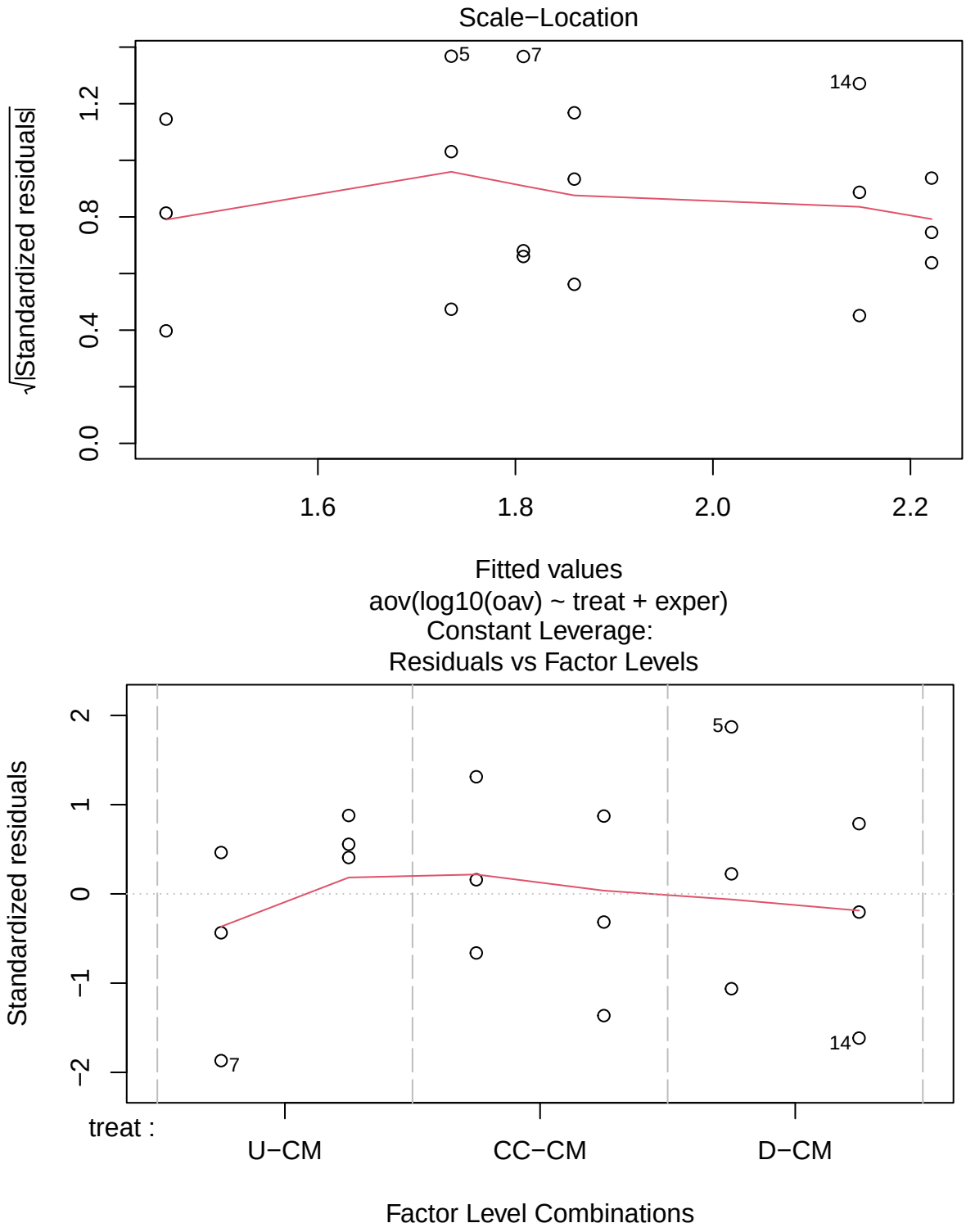
```
##                          2.5 %       97.5 %
## (Intercept)       -5498.53657 -7280.277710
## treatCC-CM           63.29599    48.515011
## treatD-CM            28.63601    -0.102811
## experExperiment 2  -125.61220  -197.413562
```

Use this model in paper. Back-transformed coef and confint results give % reduction relative to reference. CC-CM clearly lower, D-CM not.

Check diagnostic plots.

```
plot(modinit2, ask = FALSE)
```


Residuals vs Fitted
Residuals
0.10
0.05
0.00
−0.10
5
7
14
1.6
1.8
2.0
2.2
Fitted values
aov(log10(oav) ~ treat + exper)
Normal Q−Q
Standardized residuals
2
1
0
−1
−2
5
14
7
−2
−1
0
1
2
Theoretical Quantiles
aov(log10(oav) ~ treat + exper)



These look good.