```
> library(Stat5303libs);library(cfcdae);library(lme4)

> gums <- read.table("gum.dat.txt",header=TRUE);gums
```

More emulsion data. Eight samples of unprocessed gum from acacia senegal trees are obtained. Four samples come from one batch of gum, and the other four come from a second batch of gum. Within each batch, the four samples are randomly assigned to the factor/level combinations of two factors. The first factor is whether or not the gum is demineralized; the second factor is whether or not the gum is pasteurized. An emulsion is made using each sample of gum. Each emulsion is then split into three smaller parts, with the parts randomly assigned to be pH adjusted to 2.5, 4.5, or 5.5 using citric acid. The measured response is the time (in hours) till the emulsion fails, called the breakage time.

```
       y past demin ph block
1  198.5    2     2  1     1
2  299.0    2     2  2     1
3  223.1    2     2  3     1
4  166.6    1     1  1     1
5  196.5    1     1  2     1
6  178.9    1     1  3     1
7  160.7    2     1  1     1
8  151.1    2     1  2     1
9  146.5    2     1  3     1
10 146.3    1     2  1     1
11 169.3    1     2  2     1
12 198.1    1     2  3     1
13 236.3    2     2  1     2
14 330.7    2     2  2     2
15 281.2    2     2  3     2
16 151.8    1     1  1     2
17 156.0    1     1  2     2
18 159.7    1     1  3     2
19 171.8    2     1  1     2
20 159.3    2     1  2     2
21 155.9    2     1  3     2
22 175.2    1     2  1     2
23 182.2    1     2  2     2
24 136.2    1     2  3     2
> gums <- within(gums,{past<-factor(past);demin<-factor(demin);
       ph<-factor(ph);block<-factor(block)})
> fit1 <- lmer(y~block+ph*past*demin+(1|block:demin:past),data=gums)
```

This is a split plot, with batch as block, sample as whole plot, and part of the emulsion as split plot. Blocks and whole plot treatments together enumerate all whole plots, so we need a random effect enumerated by all of the block by demineralization by pasteurization combinations.

```
> wplots <- with(gums,join(block,demin,past));wplots
```

Alternatively, we can make a new variable to enumerate whole plots.

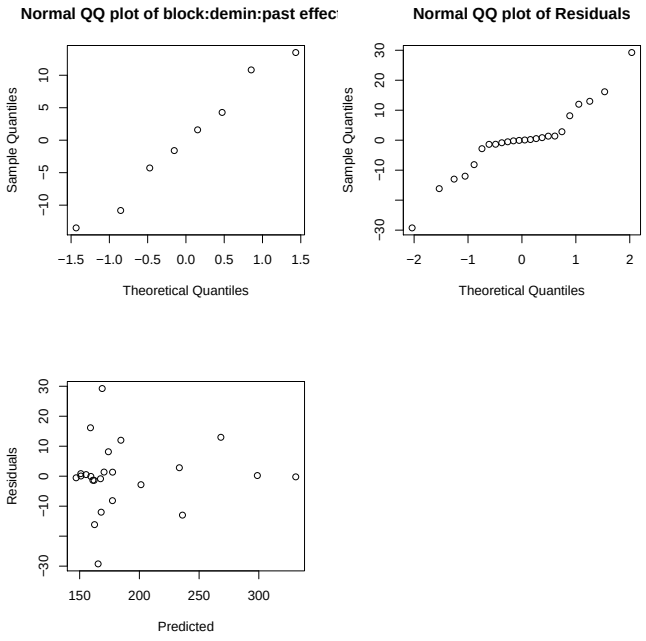
```
 [1] 1:2:2 1:2:2 1:2:2 1:1:1 1:1:1 1:1:1 1:1:2 1:1:2 1:1:2 1:2:1 1:2:1 1:2:1 2:2:2
[14] 2:2:2 2:2:2 2:1:1 2:1:1 2:1:1 2:1:2 2:1:2 2:1:2 2:2:1 2:2:1 2:2:1
Levels: 1:1:1 1:1:2 1:2:1 1:2:2 2:1:1 2:1:2 2:2:1 2:2:2

> fit1b <- lmer(y~block+ph*past*demin+(1|wplots),data=gums)
```

This will give the same as fit1.

> **`lmer.plot(fit1)`**

Looking at the residual plots we find long tails and much wider variance at low predicted values.

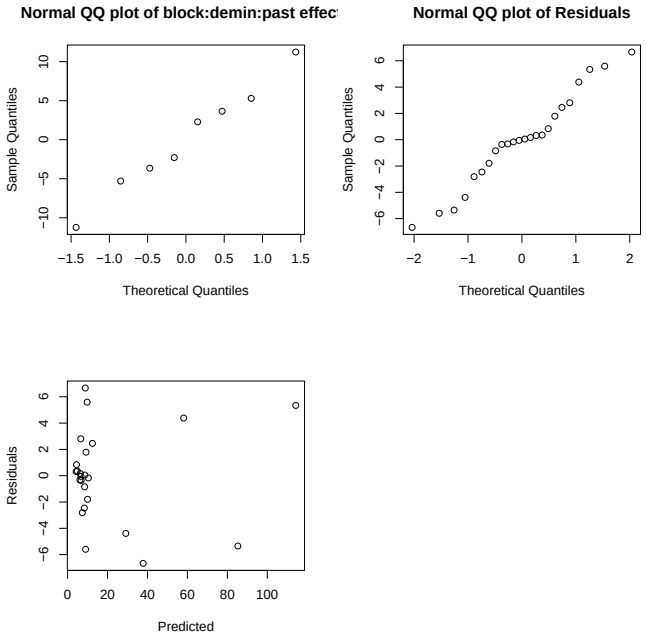


> **`fit2 <- lmer((y/100)^4~block+ph*past*demin+(1|block:demin:past),data=gums)`**

After some trial and error, the fourth power seems to make residuals plots look better (the division by 100 just retains a more comprehensible magnitude). However, I am unconvinced by a fourth power transformation.

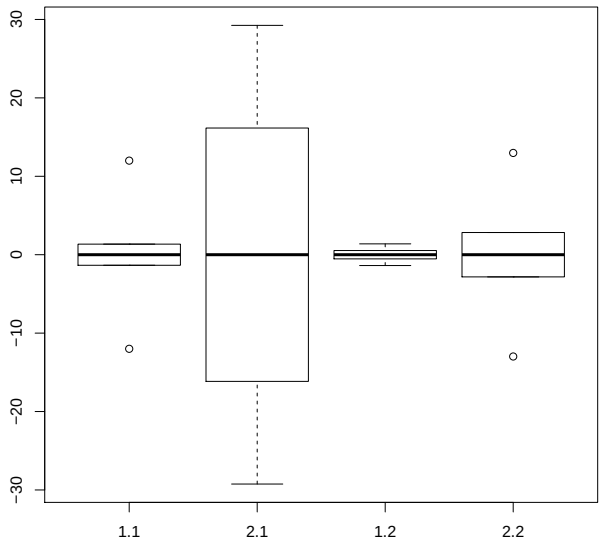
> **`lmer.plot(fit2)`**

A little better, but still not great.



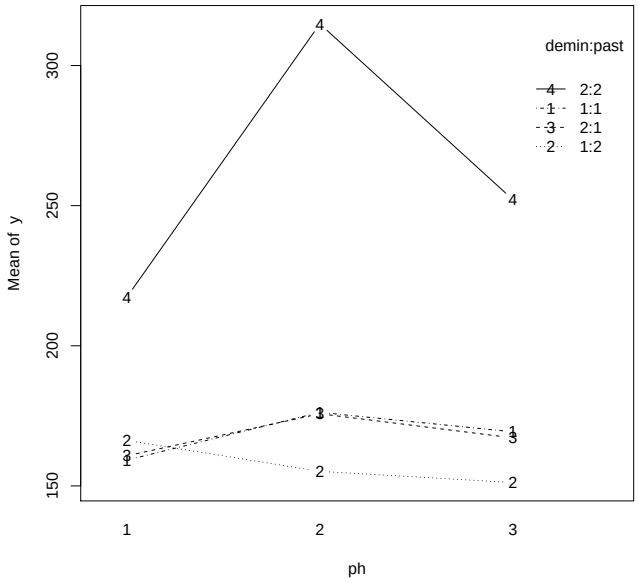
> **`boxplot(residuals(fit1)~gums$demin+gums$past)`**

After a little detective work, what we find is that there is more residual variance at demin level 2 (use demineralization), especially at past level 1 (no Pasteurization).



```
> with(gums,interactplot(ph,demin:past,y))
```

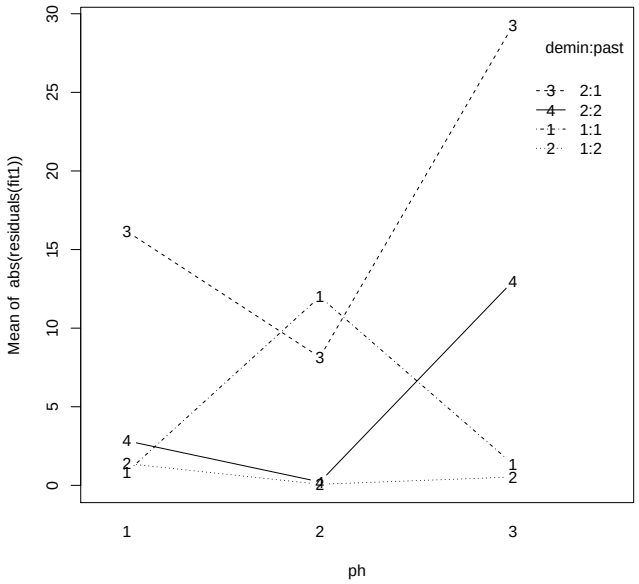
Treatment means.




```
> with(gums,interactplot(ph,demin:past,abs(residuals(fit1))))
```

Treatment absolute residuals.

What we see is that the size of the residuals depends on factor/level combination, but not so much on mean response. This means that power transformations are not going to be terribly effective, as we have seen.



```
> #
```

What is this going to mean? Some contrasts between pH levels within certain demin:past combinations will have their variance underestimated (e.g. the 2:1 combination), and in other combination the variance will be overestimated (e.g., the 1:2 combination).
For now we will go ahead using fit2; later we may explore some other things we might do.

```
> summary(fit2)
```

Block variance is estimated to be fairly large on this scale (it is only about the size of σ^2 on the original scale).
We can clearly see that the comparisons at the whole plot level have more variability than those at the split plot level (among pH levels).

```
Linear mixed model fit by REML ['lmerMod']
Formula: (y/100)^4 ~ block + ph * past * demin + (1 | block:demin:past)
   Data: gums

REML criterion at convergence: 114.5

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.1375 -0.3339  0.0000  0.3339  1.1375

Random effects:
 Groups           Name        Variance Std.Dev.
 block:demin:past (Intercept) 125.64   11.209
 Residual                      34.32    5.859
Number of obs: 24, groups:  block:demin:past, 8

Fixed effects:
                 Estimate Std. Error t value
(Intercept)        19.740      4.140   4.769
block1             -3.259      4.140  -0.787
ph1                -8.602      1.691  -5.086
ph2                11.666      1.691   6.898
past1             -11.183      4.140  -2.702
demin1            -12.395      4.140  -2.994
ph1:past1           6.800      1.691   4.021
ph2:past1         -10.206      1.691  -6.034
ph1:demin1          8.356      1.691   4.940
ph2:demin1        -10.890      1.691  -6.439
past1:demin1       12.271      4.140   2.964
ph1:past1:demin1   -8.479      1.691  -5.013
ph2:past1:demin1   11.413      1.691   6.748

. . .
```

> **Anova(fit2,test="F")**

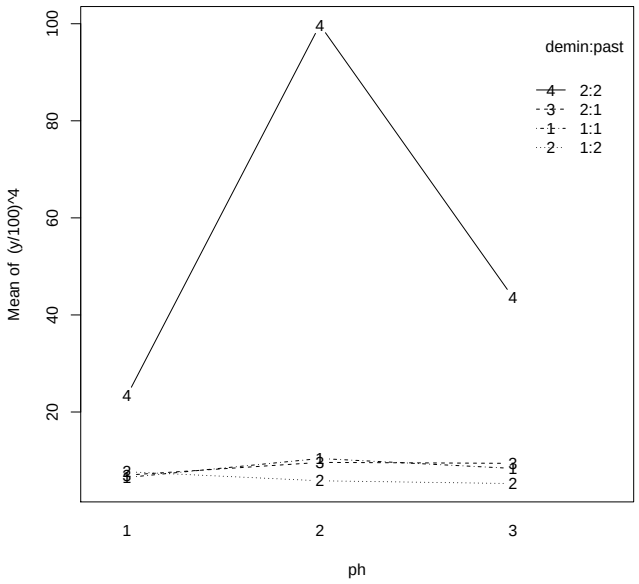
Most terms are at least marginally significant, and anything with pH is very significant.

```
Analysis of Deviance Table (Type II Wald F tests with Kenward-Roger df)

Response: (y/100)^4
                    F Df Df.res    Pr(>F)
block          0.6199  1      3 0.4885699
ph            25.5774  2      8 0.0003345 ***
past           7.2988  1      3 0.0736845 .
demin          8.9653  1      3 0.0579355 .
ph:past       18.8775  2      8 0.0009346 ***
ph:demin      22.7052  2      8 0.0005033 ***
past:demin     8.7869  1      3 0.0593402 .
ph:past:demin 24.5609  2      8 0.0003847 ***
```

> **with(gums,interactplot(ph,demin:past,(y/100)^4))**

Let's look at an interaction plot to try to understand all these interactions. We we see is that basically nothing happens unless both demineralization and Pasteurization are used, and in that case the values are much higher and pH has an effect.



> **fit2b <- lmer((y/100)^4~ph:past:demin-1+(1|block)+(1|block:demin:past),data=gums)**

The interaction plot suggests that treatment means will be more interpretable than the main effects and interactions. In this model we use the three factor interaction without main effects and with -1 to get the treatment means. We also use block as random. If we used block as fixed (as before), we would need block to come after ph:past:demin, but there seems to be no way to force lmer to do that.

```
> summary(fit2b)
```

Note that the treatment means have different values but all have the same fairly large SE. In this formulation, the variability of every treatment mean is affected by block variability, so variances are high. Variances of contrasts, particularly contrasts involving pH, will generally be much smaller.

```
Linear mixed model fit by REML ['lmerMod']
Formula: (y/100)^4 ~ ph:past:demin - 1 + (1 | block) + (1 | block:demin:past)
   Data: gums

REML criterion at convergence: 94.3

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.0678 -0.3572  0.0000  0.3572  1.0678

Random effects:
 Groups           Name        Variance Std.Dev.
 block:demin:past (Intercept) 112.62   10.612
 block            (Intercept)   0.00    0.000
 Residual                      34.32    5.859
Number of obs: 24, groups:  block:demin:past, 8; block, 2

Fixed effects:
                 Estimate Std. Error t value
ph1:past1:demin1    6.507      8.572   0.759
ph2:past1:demin1   10.416      8.572   1.215
ph3:past1:demin1    8.374      8.572   0.977
ph1:past2:demin1    7.690      8.572   0.897
ph2:past2:demin1    5.826      8.572   0.680
ph3:past2:demin1    5.257      8.572   0.613
ph1:past1:demin2    7.002      8.572   0.817
ph2:past1:demin2    9.618      8.572   1.122
ph3:past1:demin2    9.421      8.572   1.099
ph1:past2:demin2   23.352      8.572   2.724
ph2:past2:demin2   99.763      8.572  11.639
ph3:past2:demin2   43.650      8.572   5.092

. . .
```

```
> fit3 <- lmer((y/100)^4~block+ph*(demin+past)+(1|block:past:demin),data=gums,
         subset=!(past==2 & demin==2))
```

Let's follow up on that and refit using the data that are not Pasteurized and/or not demineralized. Note that we can only use the additive model at the whole plot level, because removing the 2,2 combination prevents us from estimating the interaction.

```
> Anova(fit3,test="F")
```

Absolutely nothing is happening outside of when both demineralization and Pasteurization are used.

```
Analysis of Deviance Table (Type II Wald F tests with Kenward-Roger df)

Response: (y/100)^4
             F Df Df.res Pr(>F)
block    0.7573  1      2 0.4759
ph       0.2285  2      6 0.8023
demin    0.0114  1      2 0.9248
past     0.8753  1      2 0.4482
ph:demin 0.0558  2      6 0.9462
ph:past  0.5601  2      6 0.5984
```

```
> gums <- within(gums,pd <- (past==2 & demin==2)*1);gums$pd
```

What I want now is a factor with four levels: three levels for the three groups with different pHs but with both demineralization and Pasteurization at the high level, and then a fourth group for everything else. Here we do this in two steps. First we identify the data with both demineralization and Pasteurization. Then we separate the selected subset according to pH values.

```
 [1] 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0
```

```
> gums <- within(gums,{pdph <- pd*as.numeric(ph); pdph <- as.factor(pdph)});gums$pdph
 [1] 1 2 3 0 0 0 0 0 0 0 0 0 1 2 3 0 0 0 0 0 0 0 0 0
Levels: 0 1 2 3
```

```
> fit4 <- lmer((y/100)^4~pdph+block-1+(1|block:past:demin),data=gums)
```

Fit a model with just the indictor for the four groups we just identified. Put pdph in first, because I want to "subtract" 1 and get means rather than effects. If we put block in first then we would get block means instead of block effects.

```
> summary(fit4)
```

The "nothing is happening" mean is 7.79, and we can see means (with SEs) of the other three combinations.

```
Linear mixed model fit by REML ['lmerMod']
Formula: (y/100)^4 ~ pdph + block - 1 + (1 | block:past:demin)
   Data: gums

REML criterion at convergence: 132.8

Scaled residuals:
     Min       1Q   Median       3Q      Max
-1.41948 -0.37913 -0.06805  0.36531  1.41948

Random effects:
 Groups           Name        Variance Std.Dev.
 block:past:demin (Intercept) 76.41    8.742
 Residual                     21.78    4.666
Number of obs: 24, groups:  block:past:demin, 8

Fixed effects:
       Estimate Std. Error t value
pdph0     7.790      3.734   2.086
pdph1    23.352      7.007   3.333
pdph2    99.763      7.007  14.238
pdph3    43.650      7.007   6.230
block1   -3.259      3.234  -1.008

. . .
```

```
> fit4.mcmc <- lmer.mcmc(fit4,50000)
```

Let's look at Bayesian inference.

```
> lmer.mcmc.intervals(fit4.mcmc)
```

MCMC thinks SEs are a little higher.

```
                     lower    median      upper         SE
pdph0             -2.047386  7.913183  20.241978   5.478459
pdph1              5.042011 22.611269  41.552386   9.064313
pdph2             81.883844 99.345227 117.781176   8.881268
pdph3             25.305345 43.305117  61.545301   8.763835
block1           -14.193284 -3.441624   6.326855   4.795280
block:past:demin  26.497674 90.110885 800.598889 237.990308
sigma2            11.488296 22.539727  57.486464  11.930594
```

```
> diff12 <- fit4.mcmc$mcmcout[,2]-fit4.mcmc$mcmcout[,1]
```

Let's look at the difference between the phph1 group and the phph0 group.

```
> mean(diff12);sd(diff12);22.61-7.91
```

In our model, results for pdph0 and pdph1 are independent (they share no common variance components) so the difference should have standard deviation $\sqrt{9.06^2 + 5.48^2} = 10.59$. In fact, the sd of the difference is very close to that, and the mean of the differences is very close to the difference of the medians, which should be true if things are normally distributed.

```
[1] 14.70794
[1] 10.66334
[1] 14.7
```

```
> sort(diff12)[c(125,5000-124)]
```

The 95% posterior interval contains zero, so we don't have evidence that they are different.

```
     6681     47641
-7.528157 36.176696
```

```
> diff24 <- fit4.mcmc$mcmcout[,4]-fit4.mcmc$mcmcout[,2]
```

As a contrast, now loot at pdph3 minus pdph1. The SDs for each of these are about 9.

```
> mean(diff24);sd(diff24);43.31-22.61
```

The mean difference still matches up well with the difference of medians (as we would hope), but the SD of the difference is much smaller than the SD of either. This is because these quantities share some variance components, which gives them positive correlation. That common variance cancels in the difference giving us this much smaller SD.

```
[1] 20.45124
[1] 5.221042
[1] 20.7
```

```
> sort(diff24)[c(125,5000-124)]
```

The posterior interval for this difference is nowhere near zero.

```
   17151     1851
10.23074 30.78660
```

```
> #
```

I now return briefly to explore what we might do to account for what appears to be unequal error variances that depend on the whole plot treatment combinations, but not so much on the means (i.e., BoxCox doesn't really fix it). This is not a terribly unusual situation, but it is still a nuisance (or, as they say, it's a bad situation, but a situation none the less).

What I am going to demonstrate here is using lme (from package nlme) along with a way to tell lme that the residual variances should be different in different groups.

```
> library(nlme)
```

```
> fit10 <- lme(y~past:demin:ph-1,random=~1|block/wplots,data=gums)
```

This is the lme version of the approach where we fit the treatment means with random blocks and whole plots.

```
> summary(fit10)
```

Block variance is nearly zero.

```
Linear mixed-effects model fit by REML
 Data: gums
       AIC     BIC    logLik
  146.5084 153.782 -58.25419

Random effects:
 Formula: ~1 | block
        (Intercept)
StdDev: 0.001097525

 Formula: ~1 | wplots %in% block
        (Intercept) Residual
StdDev:    14.36895 18.47759

Fixed effects: y ~ past:demin:ph - 1
                  Value Std.Error DF   t-value p-value
past1:demin1:ph1 159.20  16.55125  5  9.618607   2e-04
past2:demin1:ph1 166.25  16.55125  5 10.044557   2e-04
past1:demin2:ph1 160.75  16.55125  5  9.712256   2e-04
past2:demin2:ph1 217.40  16.55125  5 13.134958   0e+00
past1:demin1:ph2 176.25  16.55125  5 10.648741   1e-04
past2:demin1:ph2 155.20  16.55125  5  9.376934   2e-04
past1:demin2:ph2 175.75  16.55125  5 10.618532   1e-04
past2:demin2:ph2 314.85  16.55125  5 19.022730   0e+00
past1:demin1:ph3 169.30  16.55125  5 10.228833   2e-04
past2:demin1:ph3 151.20  16.55125  5  9.135260   3e-04
past1:demin2:ph3 167.15  16.55125  5 10.098934   2e-04
past2:demin2:ph3 252.15  16.55125  5 15.234497   0e+00

. . .
```

```
> fit10b <- lmer(y~0+past:demin:ph+(1|block)+(1|wplots),data=gums)
```

Same thing with lmer.

```
> summary(fit10b)
```

Output basically the same.

```
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ 0 + past:demin:ph + (1 | block) + (1 | wplots)
   Data: gums

REML criterion at convergence: 116.5

Scaled residuals:
    Min      1Q  Median      3Q     Max
```

```
-1.5581 -0.1703  0.0000  0.1703  1.5581

Random effects:
 Groups   Name        Variance Std.Dev.
 wplots   (Intercept) 206.5    14.37
 block    (Intercept)   0.0     0.00
 Residual             341.4    18.48
Number of obs: 24, groups:  wplots, 8; block, 2

Fixed effects:
                 Estimate Std. Error t value
past1:demin1:ph1   159.20      16.55   9.619
past2:demin1:ph1   166.25      16.55  10.045
past1:demin2:ph1   160.75      16.55   9.712
past2:demin2:ph1   217.40      16.55  13.135
past1:demin1:ph2   176.25      16.55  10.649
past2:demin1:ph2   155.20      16.55   9.377
past1:demin2:ph2   175.75      16.55  10.619
past2:demin2:ph2   314.85      16.55  19.023
past1:demin1:ph3   169.30      16.55  10.229
past2:demin1:ph3   151.20      16.55   9.135
past1:demin2:ph3   167.15      16.55  10.099
past2:demin2:ph3   252.15      16.55  15.234
```

```
> fit11 <- lme(y~past:demin:ph-1,random=~1|block/wplots,data=gums,
      weights=varIdent(form=~1|past*demin))
```

Now the new bit. The weights=varIdent argument tells lme to fit a different residual variance for each past by demin combination and then form appropriate weightings based on those variances.

```
> summary(fit11)
```

What we see is that the 1:1 combination is taken as a standard, the 2:1 combination has a smaller variance, the 1:2 combination has a larger variance, and the 2:2 is like 1:1.
One result of this is that different means have different standard errors, unlike what we saw above.

```
Linear mixed-effects model fit by REML
 Data: gums
       AIC     BIC    logLik
  138.8587 147.587 -51.42933

Random effects:
 Formula: ~1 | block
        (Intercept)
StdDev: 0.001151596

 Formula: ~1 | wplots %in% block
        (Intercept) Residual
StdDev:    17.73408 9.695371
```

```
Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | past * demin
 Parameter estimates:
     1*1      2*1      1*2      2*2
1.0000000 0.1062495 3.1726665 1.0502877
Fixed effects: y ~ past:demin:ph - 1
                  Value Std.Error DF   t-value p-value
past1:demin1:ph1 159.20  14.29157  5 11.139435  0.0001
past2:demin1:ph1 166.25  12.56103  5 13.235383  0.0000
past1:demin2:ph1 160.75  25.10664  5  6.402690  0.0014
past2:demin2:ph1 217.40  14.46011  5 15.034461  0.0000
past1:demin1:ph2 176.25  14.29157  5 12.332446  0.0001
past2:demin1:ph2 155.20  12.56103  5 12.355677  0.0001
past1:demin2:ph2 175.75  25.10664  5  7.000141  0.0009
past2:demin2:ph2 314.85  14.46011  5 21.773690  0.0000
past1:demin1:ph3 169.30  14.29157  5 11.846145  0.0001
past2:demin1:ph3 151.20  12.56103  5 12.037232  0.0001
past1:demin2:ph3 167.15  25.10664  5  6.657603  0.0012
past2:demin2:ph3 252.15  14.46011  5 17.437624  0.0000
```

```
> fit12 <- lme(y~past*demin*ph,random=~1|block/wplots,data=gums,
   weights=varIdent(form=~1|past*demin))
```

OK, let's now use the normal factors.

```
> anova(fit12)
```

Here we have an anova that takes into account the different variances.

```
              numDF denDF  F-value p-value
(Intercept)       1     8 762.3425  <.0001
past              1     3   7.7087  0.0692
demin             1     3  19.2655  0.0219
ph                2     8 101.8683  <.0001
past:demin        1     3  13.2651  0.0357
past:ph           2     8   5.0695  0.0378
demin:ph          2     8  50.9841  <.0001
past:demin:ph     2     8   5.3389  0.0337
```

```
> Anova(fit1,test="F")
```

Compare with the model assuming constant variance. Some pH terms are much less significant in the unmodified approach.

```
Analysis of Deviance Table (Type II Wald F tests with Kenward-Roger df)

Response: y
                    F Df Df.res  Pr(>F)
block          0.1291  1      3 0.74313
ph             5.3956  2      8 0.03285 *
past           8.3897  1      3 0.06268 .
demin         13.0953  1      3 0.03628 *
ph:past        1.3621  2      8 0.30968
ph:demin       4.1735  2      8 0.05736 .
```

```
past:demin    13.2814  1      3 0.03563 *
ph:past:demin  4.4760  2      8 0.04960 *
```

```
> #
```

Next is Example 16.7 of the text. Tables are blocks, nitrogen is the whole plot treatment factor; wetlands are split plots, and weed seeding is the split plot treatment factor; sections of wetlands (trays) are split split plots, and clipping is the split split plot treatment factor.

```
> weeds <-read.table("http://www.stat.umn.edu/~gary/book/fcdae.data/exmpl16.7",
+ header=TRUE)

> weeds <- within(weeds,table <- factor(table);nitrogen<-factor(nitrogen);
    weed <- factor(weed);clipping<-factor(clipping);wetland<-factor(wetland))

> weeds
   table nitrogen weed clipping tray wetland pct.nonweed.biomass
1      1        1    1        1    1       1                 87.2
2      1        1    1        2    1       1                 88.8
3      1        1    2        1    1       2                 70.4
4      1        1    2        2    1       2                 75.7
5      1        1    3        1    1       3                 75.9
...
46     2        4    2        2    8      23                 43.9
47     2        4    3        1    8      24                 41.9
48     2        4    3        2    8      24                 45.1

> out1 <- lme(pct.nonweed.biomass ~ table + nitrogen*weed*clipping,
   random=~1|tray/wetland,data=weeds)
```

This is the basic split-split plot analysis. We can use the nesting notation in the random part because wetlands *are* nested in trays.

> **summary(out1)**

We can see a big tray random effect, and a wetland effect that is larger than the residual noise. This also shows up in the different sizes of the SEs for estimates at different levels. Notice that non-weed biomass decreases as nitrogen increases.

```
Linear mixed-effects model fit by REML
 Data: NULL
       AIC      BIC    logLik
  221.7445 253.5384 -82.87227

Random effects:
 Formula: ~1 | tray
        (Intercept)
StdDev:    3.800906

 Formula: ~1 | wetland %in% tray
        (Intercept) Residual
StdDev:    1.617708 1.033602

Fixed effects: pct.nonweed.biomass ~ table + nitrogen * weed * clipping
                            Value Std.Error DF   t-value p-value
(Intercept)               64.23333 1.3918184 12  46.15066  0.0000
table1                     0.54583 1.3918184  3   0.39217  0.7211
nitrogen1                 10.98333 2.4107002  3   4.55608  0.0198
nitrogen2                  3.35833 2.4107002  3   1.39309  0.2579
nitrogen3                 -3.19167 2.4107002  3  -1.32396  0.2774
weed1                     17.04792 0.5124407  8  33.26808  0.0000
weed2                     -9.42708 0.5124407  8 -18.39644  0.0000
clipping1                 -1.61667 0.1491876 12 -10.83647  0.0000
nitrogen1:weed1           -8.58958 0.8875734  8  -9.67760  0.0000
nitrogen2:weed1           -2.31458 0.8875734  8  -2.60777  0.0312
nitrogen3:weed1            2.68542 0.8875734  8   3.02557  0.0164
nitrogen1:weed2            4.08542 0.8875734  8   4.60291  0.0017
nitrogen2:weed2            1.76042 0.8875734  8   1.98340  0.0826
nitrogen3:weed2           -1.16458 0.8875734  8  -1.31210  0.2259
nitrogen1:clipping1        0.08333 0.2584005 12   0.32250  0.7526
nitrogen2:clipping1       -0.12500 0.2584005 12  -0.48375  0.6373
nitrogen3:clipping1        0.15833 0.2584005 12   0.61274  0.5515
weed1:clipping1            0.09792 0.2109831 12   0.46410  0.6509
weed2:clipping1           -0.02708 0.2109831 12  -0.12837  0.9000
nitrogen1:weed1:clipping1  0.46042 0.3654335 12   1.25992  0.2317
nitrogen2:weed1:clipping1 -0.53125 0.3654335 12  -1.45375  0.1717
nitrogen3:weed1:clipping1  0.18542 0.3654335 12   0.50739  0.6211
nitrogen1:weed2:clipping1 -0.56458 0.3654335 12  -1.54497  0.1483
nitrogen2:weed2:clipping1  0.24375 0.3654335 12   0.66702  0.5174
nitrogen3:weed2:clipping1  0.18542 0.3654335 12   0.50739  0.6211
...

Number of Observations: 48
Number of Groups:
             tray wetland %in% tray
                8                24
```

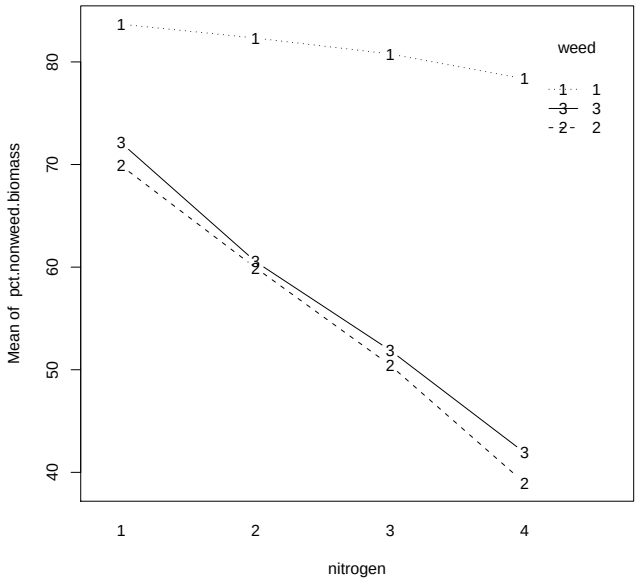
> **anova(out1)**

For the split-split plot, the anova for lme() works fine. All main effects are significant, plus the nitrogen by weed interaction.

```
                      numDF denDF   F-value p-value
(Intercept)               1    12 2129.8830  <.0001
table                     1     3    0.1538  0.7211
nitrogen                  3     3   11.4610  0.0377
weed                      2     8  555.4531  <.0001
clipping                  1    12  117.4290  <.0001
nitrogen:weed             6     8   24.5814  0.0001
nitrogen:clipping         3    12    0.2293  0.8742
weed:clipping             2    12    0.1149  0.8925
nitrogen:weed:clipping    6    12    0.7514  0.6203
```

> **interactplot(nitrogen,weed,pct.nonweed.biomass)**

Nonweed biomass decreases as nitrogen increases, but the decrease is much larger for the weed-seeded treatments.



> **emulsions <- read.table("emul2.dat.txt",header=TRUE)**

More emulsion data. Three emulsifiers: milk protein concentrate (MPC), modified milk protein concentrate (mMPC), and sodium caseinate (NC). An emulsion is made from each of the emulsifiers and the turbidity is measured at 0, 1, 2, 3, 4, 5, 6, 12, 24, 36, and 48 hours. A good emulsifier should keep the turbidity high. The experiment is then repeated two more times.

This is a repeated measures design, because we cannot randomize time.

```
> emulsions
       y protein time block
1  1.035       1    1     1
2  1.051       1    1     2
3  1.054       1    1     3
4  1.047       2    1     1
5  1.067       2    1     2
6  1.087       2    1     3
7  1.036       3    1     1
8  1.078       3    1     2
9  1.083       3    1     3
10 1.032       1    2     1
11 1.044       1    2     2
12 1.016       1    2     3
13 1.091       2    2     1
14 1.067       2    2     2
15 1.058       2    2     3
16 1.087       3    2     1
17 1.071       3    2     2
18 1.079       3    2     3
19 0.978       1    3     1
20 1.016       1    3     2
21 0.978       1    3     3
22 1.085       2    3     1
23 1.072       2    3     2
24 1.060       2    3     3
25 1.080       3    3     1
26 1.066       3    3     2
27 1.065       3    3     3
28 0.927       1    4     1
29 0.998       1    4     2
30 0.974       1    4     3
31 1.074       2    4     1
32 1.071       2    4     2
33 1.060       2    4     3
34 1.028       3    4     1
35 1.066       3    4     2
36 1.062       3    4     3
37 0.925       1    5     1
38 0.960       1    5     2
39 0.938       1    5     3
40 1.067       2    5     1
41 1.058       2    5     2
42 1.048       2    5     3
43 1.075       3    5     1
44 1.062       3    5     2
45 1.062       3    5     3
46 0.886       1    6     1
47 0.923       1    6     2
48 0.904       1    6     3
49 1.067       2    6     1
50 1.048       2    6     2
51 1.049       2    6     3
```

```
52 1.072       3    6     1
53 1.072       3    6     2
54 1.063       3    6     3
55 0.868       1    7     1
56 0.899       1    7     2
57 0.890       1    7     3
58 1.067       2    7     1
59 1.046       2    7     2
60 1.043       2    7     3
61 1.076       3    7     1
62 1.054       3    7     2
63 1.066       3    7     3
64 0.794       1    8     1
65 0.768       1    8     2
66 0.792       1    8     3
67 1.064       2    8     1
68 1.038       2    8     2
69 1.038       2    8     3
70 1.062       3    8     1
71 1.050       3    8     2
72 1.040       3    8     3
73 0.750       1    9     1
74 0.726       1    9     2
75 0.768       1    9     3
76 1.079       2    9     1
77 1.007       2    9     2
78 0.996       2    9     3
79 1.057       3    9     1
80 1.026       3    9     2
81 0.989       3    9     3
82 0.740       1   10     1
83 0.731       1   10     2
84 0.768       1   10     3
85 1.036       2   10     1
86 0.935       2   10     2
87 0.917       2   10     3
88 1.023       3   10     1
89 0.938       3   10     2
90 0.956       3   10     3
91 0.720       1   11     1
92 0.698       1   11     2
93 0.738       1   11     3
94 0.900       2   11     1
95 0.924       2   11     2
96 0.891       2   11     3
97 0.925       3   11     1
98 0.924       3   11     2
99 0.934       3   11     3
```

```
> emulsions <- within(emulsions,hrs <- c(0,1,2,3,4,5,6,12,24,36,48)[time])
```

Make a quantitative time variable.

```
> emulsions <- within(emulsions,protein<-factor(protein);
   time<-factor(time);block<-factor(block))
```

```
> emulsions <- within(emulsions,subject <- join(protein,block))
```

The "subject" in this repeated measures design (analogous to a whole plot) is formed by the block by protein combinations.

```
> out2 <- lme(y~block+protein*time,random=~1|subject,data=emulsions)
```

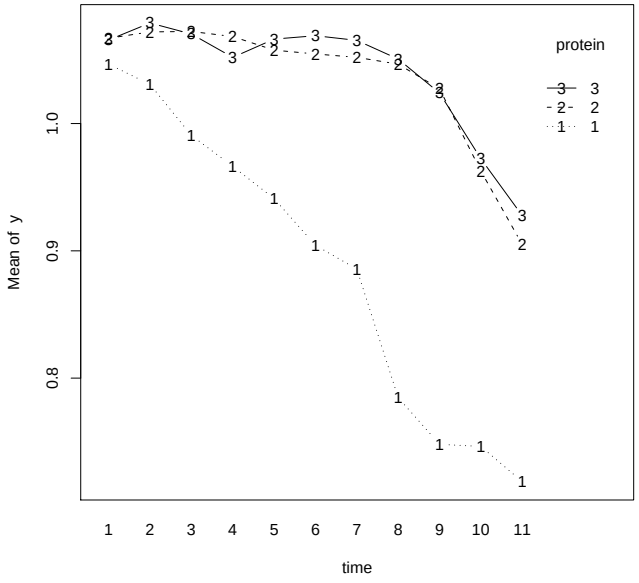
The basic univariate analysis looks just like a split plot.

```
> anova(out2)
```

Basic anova, everything is significant.

```
             numDF denDF  F-value p-value
(Intercept)      1    60 59818.30  <.0001
block            2     4     0.41  0.6911
protein          2     4   153.64  0.0002
time            10    60    99.09  <.0001
protein:time    20    60    16.97  <.0001
> interactplot(time,protein,y)
```

Protein 1 is lousy, and little difference between 2 and 3, which fail much more slowly than 1.




```
> out3 <- lme(y~block+protein*time,random=~1|subject,data=emulsions,subset=protein!=1)
```

Refit using just proteins 2 and 3.

```
> anova(out3)
```

No protein effects when we only use proteins 2 and 3.

```
             numDF denDF   F-value p-value
(Intercept)      1    40 141590.83  <.0001
block            2     2      5.15  0.1626
protein          1     2      0.87  0.4489
time            10    40     31.74  <.0001
protein:time    10    40      0.35  0.9605
```

```
> out4 <- lme(y~block+protein*poly(hrs,8),random=~1|subject,data=emulsions))
```

Now let's redo with a quantitative version of hours. We ought to be able to go to more powers, but lme is seeing singularities, so we'll stop at 8 (which is already ridiculously high).

```
> summary(out4)
```

We really only need cubic in time, but it is a different cubic for the different proteins (the interaction).

```
Linear mixed-effects model fit by REML
 Data: emulsions
       AIC      BIC   logLik
  -267.9213 -198.218 164.9607

Random effects:
 Formula: ~1 | subject
        (Intercept)   Residual
StdDev:  0.01046551 0.02023476

Fixed effects: y ~ block + protein * poly(hrs, 8)
                            Value   Std.Error DF   t-value p-value
(Intercept)             0.9876061 0.004038004 66 244.57776  0.0000
block1                  0.0049091 0.005710600  4   0.85965  0.4385
block2                 -0.0011212 0.005710600  4  -0.19634  0.8539
protein1               -0.1000606 0.005710600  4 -17.52191  0.0001
protein2                0.0474545 0.005710600  4   8.30990  0.0011
poly(hrs, 8)1          -0.6442036 0.020234758 66 -31.83649  0.0000
poly(hrs, 8)2           0.1047927 0.020234758 66   5.17884  0.0000
poly(hrs, 8)3          -0.0950574 0.020234758 66  -4.69773  0.0000
poly(hrs, 8)4           0.0300345 0.020234758 66   1.48430  0.1425
poly(hrs, 8)5           0.0240025 0.020234758 66   1.18620  0.2398
poly(hrs, 8)6          -0.0155326 0.020234758 66  -0.76762  0.4455
poly(hrs, 8)7           0.0178972 0.020234758 66   0.88448  0.3796
poly(hrs, 8)8          -0.0129973 0.020234758 66  -0.64232  0.5229
protein1:poly(hrs, 8)1 -0.3494837 0.028616270 66 -12.21276  0.0000
protein2:poly(hrs, 8)1  0.1506961 0.028616270 66   5.26610  0.0000
protein1:poly(hrs, 8)2  0.3699464 0.028616270 66  12.92784  0.0000
protein2:poly(hrs, 8)2 -0.1973916 0.028616270 66  -6.89788  0.0000
protein1:poly(hrs, 8)3 -0.1899617 0.028616270 66  -6.63824  0.0000
protein2:poly(hrs, 8)3  0.0797307 0.028616270 66   2.78620  0.0070
protein1:poly(hrs, 8)4  0.0025211 0.028616270 66   0.08810  0.9301
protein2:poly(hrs, 8)4  0.0128564 0.028616270 66   0.44927  0.6547
protein1:poly(hrs, 8)5  0.0209655 0.028616270 66   0.73264  0.4664
protein2:poly(hrs, 8)5 -0.0044158 0.028616270 66  -0.15431  0.8778
protein1:poly(hrs, 8)6 -0.0078822 0.028616270 66  -0.27544  0.7838
protein2:poly(hrs, 8)6 -0.0153755 0.028616270 66  -0.53730  0.5929
protein1:poly(hrs, 8)7 -0.0072367 0.028616270 66  -0.25289  0.8011
protein2:poly(hrs, 8)7 -0.0093175 0.028616270 66  -0.32560  0.7458
protein1:poly(hrs, 8)8  0.0043388 0.028616270 66   0.15162  0.8800
protein2:poly(hrs, 8)8  0.0209917 0.028616270 66   0.73356  0.4658
...
Number of Observations: 99
Number of Groups: 9
```

```
> out4.ar1 <- update(fit4,correlation=corAR1(form=~h2|subject))
```

OK, now here is something that we can do with lme that we cannot do with the old school analysis of repeated measures or lmer for that matter. Given that we did not randomize, we could find that there is correlation among the responses for a given subject. One concept for this correlation is that it is largest for points near in time and then decays as time gets farther apart. One potential model for this is the autoregressive model of order one (AR1). What this command does is say to keep each subject independent, but to allow for AR1 correlations between responses for a given subject, with time taken as hrs.

```
> anova(out4,out4.ar1)
```

The model with correlation is strongly preferred.

```
         Model df       AIC       BIC   logLik   Test  L.Ratio p-value
out4         1 31 -267.9213 -198.2180 164.9607
out4.ar1     2 32 -281.6305 -209.6786 172.8152 1 vs 2 15.70912   1e-04
```

```
> summary(out4.ar1)
```

If you compare the results here with the results for out4 shown just above, you see that in this case the estimated parameters haven't changed, but the standard errors of the parameters have changed, some bigger, some smaller.
Also note that the estimated correlation in the residuals at a one hour lag is .73.

```
Linear mixed-effects model fit by REML
 Data: emulsions
       AIC       BIC   logLik
  -281.6305 -209.6786 172.8152

Random effects:
 Formula: ~1 | subject
        (Intercept)   Residual
StdDev:  0.01034327 0.02436793

Correlation Structure: ARMA(1,0)
 Formula: ~hrs | subject
 Parameter estimate(s):
     Phi1
0.7301144
Fixed effects: y ~ block + protein * poly(hrs, 8)
                            Value  Std.Error DF   t-value p-value
(Intercept)             0.9876178 0.00542651 66 181.99875  0.0000
block1                  0.0111961 0.00689927  4   1.62280  0.1800
block2                 -0.0069971 0.00689927  4  -1.01418  0.3679
protein1               -0.1000188 0.00767424  4 -13.03305  0.0002
protein2                0.0474352 0.00767424  4   6.18109  0.0035
poly(hrs, 8)1          -0.6442420 0.03239667 66 -19.88606  0.0000
poly(hrs, 8)2           0.1047158 0.02684170 66   3.90124  0.0002
poly(hrs, 8)3          -0.0949096 0.02655801 66  -3.57367  0.0007
poly(hrs, 8)4           0.0299491 0.02603305 66   1.15043  0.2541
poly(hrs, 8)5           0.0239270 0.02450456 66   0.97643  0.3324
poly(hrs, 8)6          -0.0152681 0.02055279 66  -0.74287  0.4602
poly(hrs, 8)7           0.0193585 0.01510456 66   1.28164  0.2045
poly(hrs, 8)8          -0.0122608 0.01255460 66  -0.97659  0.3323
protein1:poly(hrs, 8)1 -0.3496210 0.04581581 66  -7.63101  0.0000
protein2:poly(hrs, 8)1  0.1507592 0.04581581 66   3.29055  0.0016
protein1:poly(hrs, 8)2  0.3696751 0.03795989 66   9.73857  0.0000
```

```
protein2:poly(hrs, 8)2 -0.1972652 0.03795989 66  -5.19668  0.0000
protein1:poly(hrs, 8)3 -0.1894353 0.03755870 66  -5.04371  0.0000
protein2:poly(hrs, 8)3  0.0794879 0.03755870 66   2.11637  0.0381
protein1:poly(hrs, 8)4  0.0022098 0.03681629 66   0.06002  0.9523
protein2:poly(hrs, 8)4  0.0129962 0.03681629 66   0.35300  0.7252
protein1:poly(hrs, 8)5  0.0207111 0.03465468 66   0.59764  0.5521
protein2:poly(hrs, 8)5 -0.0042907 0.03465468 66  -0.12381  0.9018
protein1:poly(hrs, 8)6 -0.0069645 0.02906604 66  -0.23961  0.8114
protein2:poly(hrs, 8)6 -0.0158117 0.02906604 66  -0.54399  0.5883
protein1:poly(hrs, 8)7 -0.0015148 0.02136107 66  -0.07092  0.9437
protein2:poly(hrs, 8)7 -0.0116788 0.02136107 66  -0.54673  0.5864
protein1:poly(hrs, 8)8  0.0032680 0.01775489 66   0.18406  0.8545
protein2:poly(hrs, 8)8  0.0195035 0.01775489 66   1.09848  0.2760
...
Standardized Within-Group Residuals:
         Min           Q1          Med           Q3          Max
-1.757836630 -0.384968555 -0.002115500  0.292879828  2.296237373

Number of Observations: 99
Number of Groups: 9
```