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

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

Data from Kirk (1982, Brooks/Cole). Adjective/noun pairs are presented on a computer screen to subjects. The adjective is present for 100ms, followed by the noun for 20 ms. If the subject does not recognize the noun, the pair is presented again with the presentation time for the noun increased by 5 ms. This is repeated until the noun is recognized. Of interest is whether the degree of association between the adjective and the noun influences how quickly the subject recognizes the noun. For example, you might recognize “ice cream” more quickly after “vanilla” than after “purple”.

Twenty-five subjects are used, and the subjects differ in their overall quickness to recognize nouns, so the subjects are blocked into five blocks of size five based on how quick they are. The experiment will be administered by five graduate students, who may have some effect, so the subjects are also blocked by which student administers the test. The response measured is the average time in ms taken to recognize 10 nouns of a given association strength. They used a Latin square because of the two blocking factors.

```
> nouns
   oquick student assocstr rtime
1       1       1        1    72
2       1       2        2    62
3       1       3        3    66
4       1       4        4    51
5       1       5        5    40
6       2       1        2    65
7       2       2        3    61
8       2       3        4    40
9       2       4        5    44
10      2       5        1    59
11      3       1        3    55
12      3       2        4    46
13      3       3        5    35
14      3       4        1    63
15      3       5        2    54
16      4       1        4    34
17      4       2        5    29
18      4       3        1    54
19      4       4        2    44
20      4       5        3    50
21      5       1        5    51
22      5       2        1    49
23      5       3        2    43
24      5       4        3    30
25      5       5        4    25

> nouns <- within(nouns,{oquick <- as.factor(oquick);student<-as.factor(student);
assocstr<-as.factor(assocstr)})
```

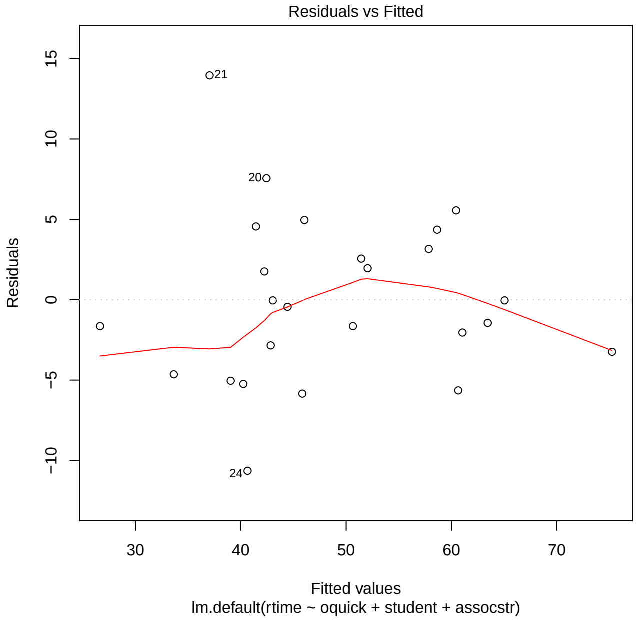
```
> tapply(as.numeric(assocstr),list(oquick,student),mean)
```

Here is the acutal LS used. Rows are blocks for overall quickness, columns are blocks for students, and matrix elements are the treatments (strength of association between adjective and noun). There's likely a cleaner, prettier way to get this table, but this works.

```
  1 2 3 4 5
1 1 2 3 4 5
2 2 3 4 5 1
3 3 4 5 1 2
4 4 5 1 2 3
5 5 1 2 3 4
```

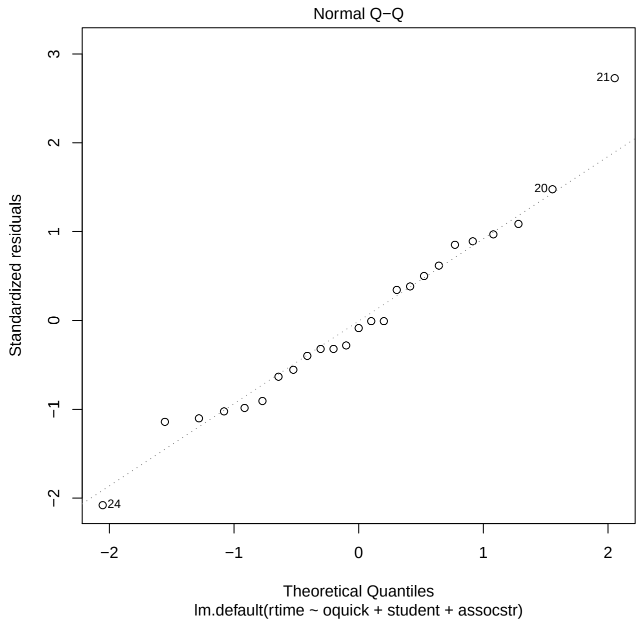
```
> fit1 <- lm(rtime˜oquick+student+assocstr,data=nouns)
> plot(fit1,which=1)
```

Look at the residuals. Looks like variance may decrease with mean?



```
> plot(fit1,which=2)
```

Normal plot shows a possible possible outlier. It does not look too bad here using standardized residuals, but as we'll see, it has a large Studentized residual.



```
> rstudent(fit1)
```

Check the Studentized residuals, and number 21 looks like an outlier.

```
           1            2            3            4            5            6
-0.616552896 -0.270301185  1.095488929  0.966564315 -1.157193872 -0.007483573
           7            8            9           10           11           12
 0.600823238 -0.538284767  0.330911261 -0.384213682 -1.113012506  0.882831558
          13           14           15           16           17           18
-1.026199973  0.841556863  0.368956162 -0.983513145 -0.899448405  0.484020874
          19           20           21           22           23           24
-0.082344464  1.563712705  4.237490113 -0.308147176 -0.007483573 -2.488739562
          25
-0.308147176
```

```
> odd <- rep(0,25);odd[21] <- 1
```

One thing that we can do with an outlier is to make a dummy variable that points just to the outlier, and then include the dummy variable in the model. Then the other data get fit as if the outlier weren't even in the data set, and the coefficient of the dummy variable tells us how far the outlier is from what would be predicted by the remainder of the data. This is a slightly more informative approach than simply deleting an outlier.

```
> fit2 <- update(fit1,~odd+.);summary(fit2)
```

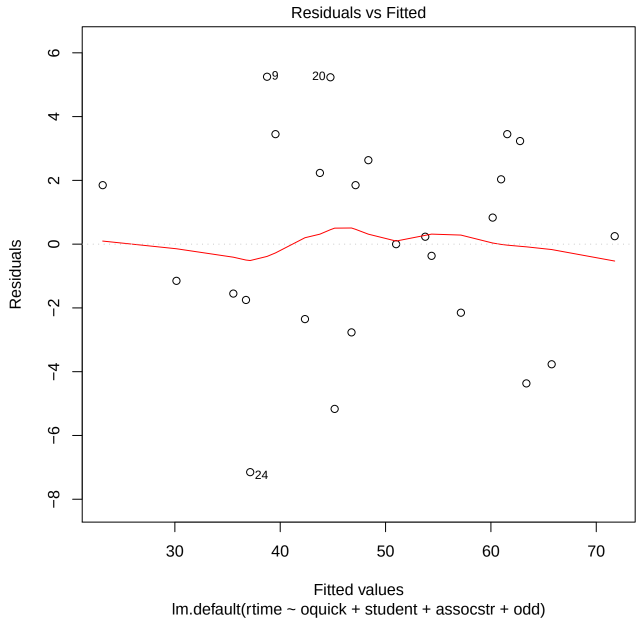
Update the model to include the dummy and look at what we got. This indicates that the outlier is about 29 ms higher than the rest of the data would have predicted. Note that the t-value for "odd" matches the studentized residual we had for response 21 in the previous model.

```
Residuals:
    Min      1Q  Median      3Q     Max
-7.1500 -2.1500  0.2333  2.2333  5.2500

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)  47.7167     0.9898  48.206 3.75e-14 ***
odd          29.0833     6.8633   4.237 0.001395 **
oquick1      10.4833     1.9217   5.455 0.000199 ***
oquick2       6.0833     1.9217   3.166 0.008990 **
oquick3       2.8833     1.9217   1.500 0.161661
oquick4      -5.5167     1.9217  -2.871 0.015223 *
student1      1.8667     2.1963   0.850 0.413483
student2      1.6833     1.9217   0.876 0.399784
student3     -0.1167     1.9217  -0.061 0.952680
student4     -1.3167     1.9217  -0.685 0.507433
assocstr1    11.6833     1.9217   6.080 7.97e-05 ***
assocstr2     5.8833     1.9217   3.061 0.010824 *
assocstr3     4.6833     1.9217   2.437 0.032996 *
assocstr4    -8.5167     1.9217  -4.432 0.001009 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 4.755 on 11 degrees of freedom
Multiple R-squared:  0.9344,Adjusted R-squared:  0.8569
F-statistic: 12.06 on 13 and 11 DF,  p-value: 0.0001094
```

```
> plot(fit2,which=1)
```

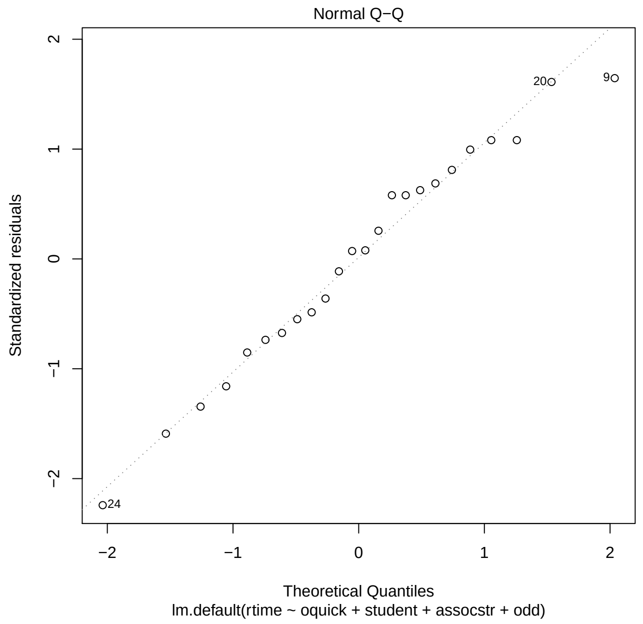
Residuals look somewhat better, although I still see a little bit of decreasing variance. Box-Cox does not suggest any transformation, however.



```
> plot(fit2,which=2)
```

Normality looks pretty good. Note that R recognizes that the outlier was fit perfectly and skips it in the normal probability plot.

```
Warning message:
Not plotting observations with leverage one:
  21
```



```
> anova(fit2)
```

Things are unbalanced and messier when we remove outliers or fit them individually. In particular, the SS and MS we get from anova() are *sequential*.

```
Analysis of Variance Table

Response: rtime
          Df  Sum Sq Mean Sq F value    Pr(>F)
odd        1    4.68    4.68  0.2071 0.6579342
oquick     4 1381.61  345.40 15.2761 0.0001827 ***
student    4  181.40   45.35  2.0057 0.1632069
assocstr   4 1976.23  494.06 21.8507 3.435e-05 ***
Residuals 11  248.72   22.61
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
> Anova(fit2)
```

We can use Type II SS. The result here for treatment is the same as in the preceding sequential anova, because treatment was entered last.

```
Anova Table (Type II tests)

Response: rtime
           Sum Sq Df F value    Pr(>F)
odd        406.00  1 17.9563  0.001395 **
oquick    1595.62  4 17.6424 9.431e-05 ***
student     57.28  4  0.6334  0.649113
assocstr  1976.23  4 21.8507 3.435e-05 ***
Residuals  248.72 11
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
> linear.contrast(fit2,assocstr,all=TRUE,jointF=TRUE)
```

Looking at the pairwise results, pairs 1 and 2, 2 and 3, and 4 and 5 cannot be distinguished from each other. This is without any multiple comparisons correction (i.e., LSD).

```
$estimates
      estimates       se   t-value      p-value   lower-ci  upper-ci
1 - 2  5.800000 3.007365 1.9285988 7.997663e-02 -0.8191651 12.419165
1 - 3  7.000000 3.007365 2.3276193 4.003671e-02  0.3808349 13.619165
1 - 4 20.200000 3.007365 6.7168442 3.300680e-05 13.5808349 26.819165
1 - 5 25.416667 3.305822 7.6884557 9.510337e-06 18.1406013 32.692732
2 - 3  1.200000 3.007365 0.3990204 6.975153e-01 -5.4191651  7.819165
2 - 4 14.400000 3.007365 4.7882453 5.638912e-04  7.7808349 21.019165
2 - 5 19.616667 3.305822 5.9339753 9.816868e-05 12.3406013 26.892732
3 - 4 13.200000 3.007365 4.3892249 1.082391e-03  6.5808349 19.819165
3 - 5 18.416667 3.305822 5.5709794 1.674154e-04 11.1406013 25.692732
4 - 5  5.216667 3.305822 1.5780240 1.428648e-01 -2.0593987 12.492732

$Ftest
        F df1 df2      p-value
 21.85073   4  11 3.434593e-05
```

```
> lines(pairwise(fit2,assocstr))
```

HSD clears things up a bit: we can't tell 4 from 5, and we can't tell 1, 2, and 3 apart.

```
5 -13.73 |
4  -8.52 |
3   4.68   |
2   5.88   |
1  11.68   |
```

```
> ((12+4)*22.611+4*398.91)/(12+4+4)
```

Here we estimate the error variance that we would have incurred had we not blocked on overall quickness. Note, I used degrees of freedom for the design without an outlier, and I have used the estimate of SS and MS for overall quickness from the type II ANOVA table.

```
(1)       97.871
```

```
> (13/15)*(19/17)*97.871/22.611
```

Here is the relative efficiency of the LS design to an RCB with only student blocks. This is very good.

```
(1)       4.193
```

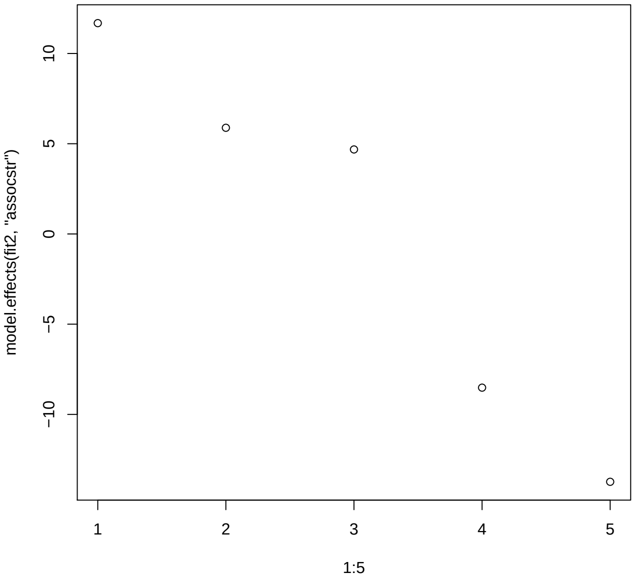
```
> model.effects(fit2,"assocstr")
```

Let's look at the treatment effects. They are basically monotone decreasing.

```
         1          2          3          4          5
 11.683333   5.883333   4.683333  -8.516667 -13.733333
```

```
> plot(1:5,model.effects(fit2,"assocstr"))
```

If we plot them against 1 through 5 (this assumes that the five treatments are in some sense equally spaced, which we don't really know for sure), then it looks pretty linear.



```
> cfs <- matrix(c(-2,-1,0,1,2,  2,-1,-2,-1,2,  -1,2,0,-2,1,  1,-4,6,-4,1),nrow=5)

> cfs
```

Linear, quadratic, cubic, and quartic contrasts.

```
     [,1] [,2] [,3] [,4]
[1,]   -2    2   -1    1
[2,]   -1   -1    2   -4
[3,]    0   -2    0    6
[4,]    1   -1   -2   -4
[5,]    2    2    1    1

> linear.contrast(fit2,assocstr,cfs,jointF=TRUE)
```

A contrast for the linear effect is highly significant, but not for the higher orders.

```
$estimates
   estimates        se    t-value      p-value   lower-ci   upper-ci
1 -65.233333  7.263476 -8.9810079 2.139925e-06 -81.220136 -49.246531
2 -10.833333  8.417040 -1.2870716 2.244931e-01 -29.359114   7.692447
3   3.383333  6.863340  0.4929573 6.317354e-01 -11.722775  18.489442
4  36.583333 17.844683  2.0500972 6.496480e-02  -2.692549  75.859215

$Ftest
        F df1 df2      p-value
 21.85073   4  11 3.434593e-05
```

```
> anova(lm(rtime˜odd+oquick+student+as.numeric(assocstr)+assocstr,data=nouns))
```

Here we fit a model with a numeric form of the treatment (just 1 through 5) and put assocstr in the model last. Since it is last in, it picks up anything nonlinear in the treatment effect. That is nonsignificant, so we would conclude that there is no evidence that the treatment effect is nonlinear.

```
Analysis of Variance Table

Response: rtime
                     Df  Sum Sq Mean Sq F value    Pr(>F)
odd                   1    4.68    4.68  0.2071 0.6579342
oquick                4 1381.61  345.40 15.2761 0.0001827 ***
student               4  181.40   45.35  2.0057 0.1632069
as.numeric(assocstr)  1 1833.84 1833.84 81.1054 2.083e-06 ***
assocstr              3  142.39   47.46  2.0992 0.1584663
Residuals            11  248.72   22.61
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
> milkdata <- read.table("milk.dat.txt",header=TRUE)
```

These data give the milk production (in pounds) for 18 cows for three consecutive five-week periods. Each cow is given a different diet in each period. The diets are good hay, poor hay, and straw. Data from John (1971 Wiley).

The experiment is arranged as a cross-over design – six Latin squares with cows as column blocks and time periods as row blocks.

```
> milkdata
   cow period square diet allcows milk
1    1      1      1    1       1  768
2    1      2      1    2       1  600
3    1      3      1    3       1  411
4    2      1      1    2       2  662
5    2      2      1    3       2  515
6    2      3      1    1       2  506
7    3      1      1    3       3  731
8    3      2      1    1       3  680
9    3      3      1    2       3  525
10   1      1      2    1       4  669
11   1      2      2    3       4  550
12   1      3      2    2       4  416
13   2      1      2    2       5  459
14   2      2      2    1       5  409
15   2      3      2    3       5  222
16   3      1      2    3       6  624
17   3      2      2    2       6  462
18   3      3      2    1       6  426
19   1      1      3    1       7 1091
20   1      2      3    2       7  798
21   1      3      3    3       7  534
22   2      1      3    2       8 1234
23   2      2      3    3       8  902
24   2      3      3    1       8  869
25   3      1      3    3       9 1300
26   3      2      3    1       9 1297
27   3      3      3    2       9  962
28   1      1      4    1      10 1105
29   1      2      4    3      10  712
```

```
30   1      3      4    2      10  453
31   2      1      4    2      11  891
32   2      2      4    1      11  830
33   2      3      4    3      11  629
34   3      1      4    3      12  859
35   3      2      4    2      12  617
36   3      3      4    1      12  597
37   1      1      5    1      13  941
38   1      2      5    2      13  718
39   1      3      5    3      13  548
40   2      1      5    2      14  794
41   2      2      5    3      14  603
42   2      3      5    1      14  613
43   3      1      5    3      15  779
44   3      2      5    1      15  718
45   3      3      5    2      15  515
46   1      1      6    1      16  933
47   1      2      6    3      16  658
48   1      3      6    2      16  576
49   2      1      6    2      17  724
50   2      2      6    1      17  649
51   2      3      6    3      17  496
52   3      1      6    3      18  749
53   3      2      6    2      18  594
54   3      3      6    1      18  612
```

```
> milkdata <- within(milkdata,{cow <- as.factor(cow);allcows <- as.factor(allcows);
 period <- as.factor(period);square <- as.factor(square);diet <- as.factor(diet)})
```

```
> fit2 <- lm(milk~allcows+period+diet,data=milkdata)
```

Fit the basic model. allcows enumerates all 18 cows, so that is our column factor.

```
> anova(fit2)
Analysis of Variance Table

Response: milk
          Df  Sum Sq Mean Sq F value    Pr(>F)
allcows   17 1710775  100634  29.127 9.384e-15 ***
period     2  814222  407111 117.833 1.742e-15 ***
diet       2  121147   60573  17.532 7.220e-06 ***
Residuals 32  110560    3455
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
> fit1 <- lm(milk~square/cow+period+diet,data=milkdata)
```

Here is an equivalent model another way. Here we have divided the between cows variation into between square, and between cows within square. They have the same 17 degree of freedom total for their SS. Note that square:cow is coming after diet (treatment) in the anova(), even through square:cow is a block difference. If we want to maintain order, for example, in the case of unbalanced data, we need to use terms() with keep.order=TRUE.

```
> anova(fit1)
Analysis of Variance Table

Response: milk
           Df  Sum Sq Mean Sq  F value    Pr(>F)    
square      5 1392534  278507  80.6099 < 2.2e-16 ***
period      2  814222  407111 117.8326 1.742e-15 ***
diet        2  121147   60573  17.5321 7.220e-06 ***
square:cow 12  318241   26520   7.6759 1.941e-06 ***
Residuals  32  110560    3455                       
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

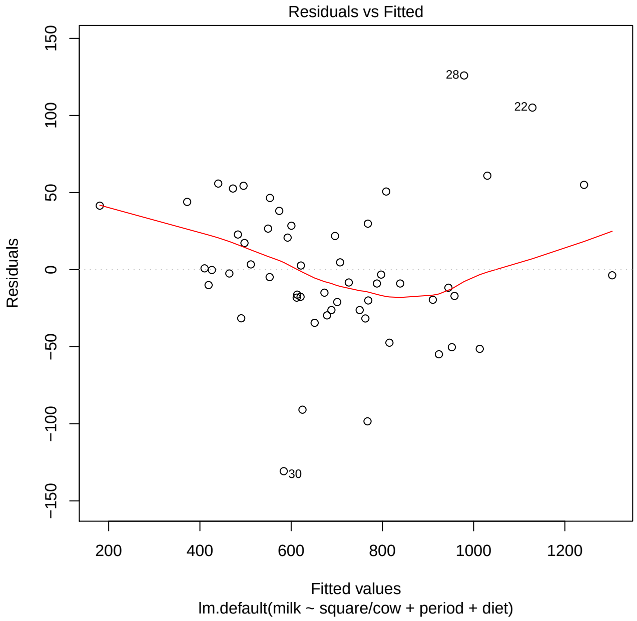
```
> 1392534+318241
```

SS for square and cow within square add up to SS for allcows in the previous model.

```
[1] 1710775
```

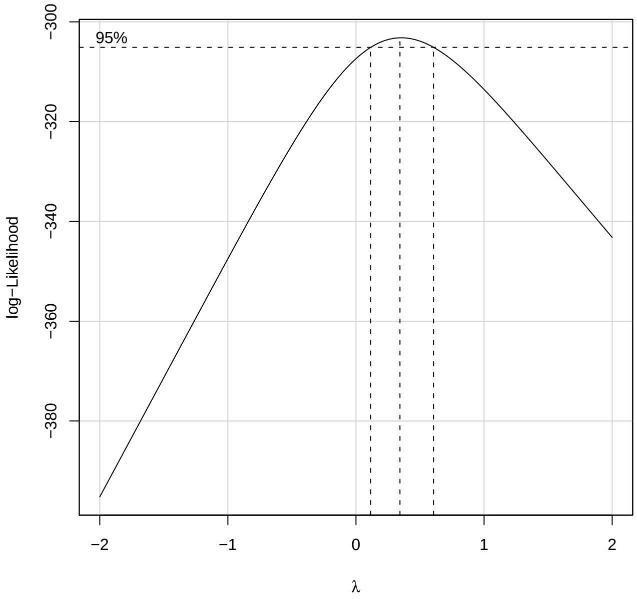
```
> plot(fit1,which=1)
```

However, we did all that work before checking residuals, and now we have the flopping fish, so we really need to do a transformation.



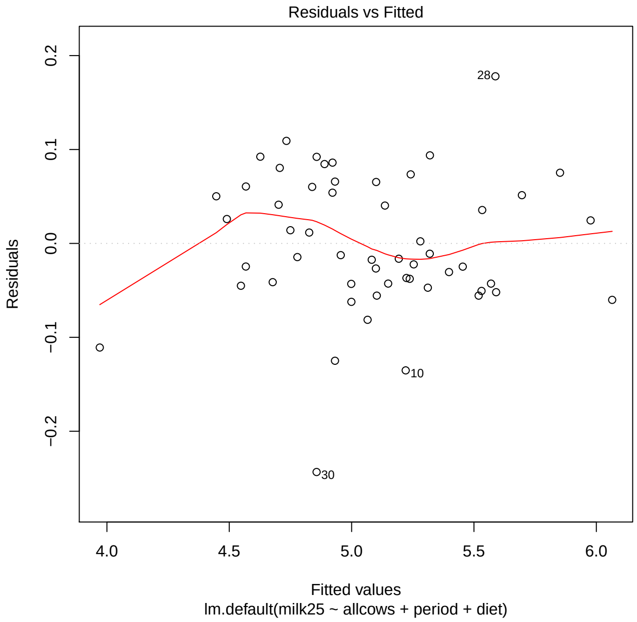
```
> boxCox(fit1)
```

Box Cox suggest something on the order of power .3



```
> fit3 <- lm(milk^.25~allcows+period+diet,data=milkdata)
> plot(fit3,which=1)
```

Now it looks like we could have an outlier.



```
> rstudent(fit3)
```

Yes, it looks like observation 30 is an outlier.

```
          1           2           3           4           5           6           7
-0.64692010  1.29061190 -0.61845953 -0.36470412 -0.19829362  0.56497868 -0.51603149
. . .
         29          30          31          32          33          34          35
 0.90391154 -4.13415097 -0.76515328 -0.41671444  1.20055351  1.31414506 -1.13157533
. . .
```

```
> odd30 <- rep(0,54);odd30[30] <- 1
```

Set up a dummy variable.

```
> fit4 <- update(fit3,~odd30+.)
```

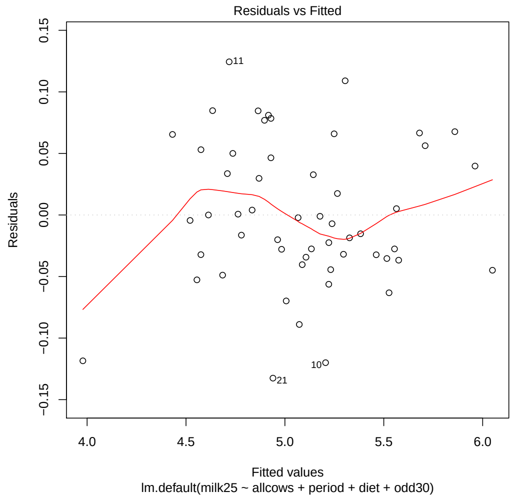
Add the dummy variable indicating the outlier to the model.

```
> rstudent(fit4)
```

OK, no additional outliers. Values not shown.

```
> plot(fit4,which=1)
```

That looks better, although there is one value sitting all by itself that might bear some scrutiny.



```
> summary(fit4)
```

OK, this model seems to fit pretty well. Diet 1 is better than diet 2, and diet 3 must be worse still. Milk production is consistently decreasing over time, so that was a useful block. The outlier is about .41 low (one the .25 scale), which is a long way from where it should be. Number 30 is cow 10, period 3, treatment 2. We need to check this to see if there is some reason that it is different.

```
Call:
lm.default(formula = milk25 ~ allcows + period + diet + odd30)

Residuals:
      Min        1Q    Median        3Q       Max
-0.132524 -0.035098 -0.005765  0.049114  0.124407
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.09285    0.01057 481.862  < 2e-16 ***
allcows1    -0.18749    0.04295  -4.365 0.000131 ***
allcows2    -0.23318    0.04295  -5.429 6.28e-06 ***
...
period1      0.27635    0.01483  18.630  < 2e-16 ***
period2     -0.01140    0.01483  -0.769 0.447814
diet1        0.11449    0.01483   7.718 1.05e-08 ***
diet2       -0.02932    0.01517  -1.933 0.062476 .
odd30       -0.41073    0.09935  -4.134 0.000251 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.07648 on 31 degrees of freedom
Multiple R-squared: 0.9801,Adjusted R-squared: 0.966
F-statistic: 69.46 on 22 and 31 DF,  p-value: < 2.2e-16

> model.effects(fit4,"diet")
          1           2           3
 0.11448965 -0.02931986 -0.08516979

> linear.contrast(fit4,diet,all=TRUE)
```

Treatment 1 (good hay) is far above treatments 2 and 3, which are just barely distinguishable from each other.

```
       estimates         se  t-value      p-value    lower-ci  upper-ci
1 - 2 0.14380950 0.02608430 5.513260 4.929294e-06 0.090610227 0.1970088
1 - 3 0.19965943 0.02549363 7.831737 7.726867e-09 0.147664827 0.2516540
2 - 3 0.05584993 0.02608430 2.141132 4.023466e-02 0.002650656 0.1090492

> Anova(fit4)
```

Let's estimate relative efficiency, so we need to get estimates adjusted for other terms.

```
Anova Table (Type II tests)

Response: milk25
          Sum Sq Df F value    Pr(>F)
odd30     0.1000  1  17.091 0.0002513 ***
allcows   5.6374 17  56.692 < 2.2e-16 ***
period    2.5290  2 216.181 < 2.2e-16 ***
diet      0.3806  2  32.535 2.432e-08 ***
Residuals 0.1813 31

> ((31+2)*(.1813/31)+2.5290)/(31+2+2)
```

Estimate the error variance if we hadn't blocked on period.

```
[1] 0.07777134

> .0778/.00585
```

This is 13 times the error variance we did have, so blocking was very beneficial!

```
[1] 13.29915
```

```
> fit5 <- lmer(milkˆ.25˜period+diet+odd30+(1|allcows),data=milkdata)
```

Try it again thinking of cows as random; that is, these cows are a sample of all cows and we're drawing inference to all cows.

```
> fit5
```

Fixed effects are estimated just like in our previous model. Cow variance is estimated at .11, or a standard deviation of .33, which is huge relative to the other effects we're looking at.

```
Linear mixed model fit by REML ['lmerMod']
Formula: milkˆ0.25 ˜ period + diet + odd30 + (1 | allcows)
   Data: milkdata

REML criterion at convergence: -24.5

Random effects:
 Groups   Name        Variance Std.Dev.
 allcows  (Intercept) 0.109690 0.33119
 Residual             0.005848 0.07647
Number of obs: 54, groups:  allcows, 18

Fixed effects:
            Estimate Std. Error t value
(Intercept)  5.09273    0.07878   64.65
period1      0.27647    0.01483   18.64
period2     -0.01128    0.01483   -0.76
diet1        0.11461    0.01483    7.73
diet2       -0.02956    0.01517   -1.95
odd30       -0.40415    0.09889   -4.09
```

```
> lmer.KR.anova(fit5)
```

In this case, results for fixed effects are a close match to doing everything as a fixed effect.

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

Response: milkˆ0.25
             F Df Df.res    Pr(>F)
period 216.549  2 31.017 < 2.2e-16 ***
diet    32.572  2 31.017 2.394e-08 ***
odd30   16.648  1 31.573 0.0002847 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
> fit6 <- lmer(milkˆ.25˜period+diet+odd30+(1|square)+(1|allcows),data=milkdata)
```

Another possibility is to model a random effect for square and then a random effect for cow within square. This would be appropriate if the squares were randomly chosen herds, and then the cows were chosen randomly from within each herd. This issue is that we might see more variability between herds than between cows within herd.

```
> summary(fit6)
```

Well, well, it looks like there could be square to square variability.

```
Linear mixed model fit by REML ['lmerMod']
Formula: milk^0.25 ~ period + diet + odd30 + (1 | square) + (1 | allcows)
   Data: milkdata

REML criterion at convergence: -35.2

Random effects:
 Groups   Name        Variance Std.Dev.
 allcows  (Intercept) 0.028088 0.16759
 square   (Intercept) 0.092446 0.30405
 Residual             0.005847 0.07646
Number of obs: 54, groups:  allcows, 18; square, 6

Fixed effects:
            Estimate Std. Error t value
(Intercept)  5.09266    0.13069   38.97
period1      0.27654    0.01483   18.65
period2     -0.01122    0.01483   -0.76
diet1        0.11468    0.01483    7.73
diet2       -0.02970    0.01516   -1.96
odd30       -0.40055    0.09809   -4.08
```

```
> AIC(fit5,fit6)
```

AIC prefers the model with square effects.

```
     df        AIC
fit5  8  -8.533075
fit6  9 -17.216903
```

```
> fit7 <- lmer(milk^.25~period+diet+odd30+(1|square),data=milkdata)
```

Fit with square only.

```
> exactRLRT(fit7,fit6,fit5)
```

Square if very significant.

```
simulated finite sample distribution of RLRT.

(p-value based on 10000 simulated values)

data:
RLRT = 10.6838, p-value = 9e-04
```

```
> fit6.mcmc <- lmer.mcmc(fit6,20000)

> lmer.mcmc.intervals(fit6.mcmc)
```

Interval estimates for variance components (and fixed effects).

```
                   lower       median        upper           SE
(Intercept)  4.808954407  5.092012486  5.397687414 0.147200639
period1      0.243562292  0.278054596  0.309481674 0.016462482
period2     -0.042089263 -0.011569385  0.019487441 0.015333610
diet1        0.084560697  0.114072309  0.145511699 0.015487300
diet2       -0.060720523 -0.029782972  0.001187014 0.015941976
odd30       -0.611429434 -0.401122401 -0.208841885 0.104102794
allcows      0.013632268  0.031228060  0.083301194 0.019311272
square       0.028051404  0.105584675  0.761573912 0.236000140
sigma2       0.003671444  0.005896938  0.010340862 0.001699295
```