```
> drill <- read.table("drill.data",header=TRUE);drill
```

These data are from Daniel (1976). The experiment was on a stone drill. Factor A is the load on the drill; B is the flow rate through the drill; C is the rotational speed; D is the type of mud used. The response is the rate of drill advance. There was just one replication. This is a 2^4 design, and the data are in standard order.

```
   load flow speed mud advance.rate
1     1    1     1   1         1.68
2     2    1     1   1         1.98
3     1    2     1   1         3.28
4     2    2     1   1         3.44
5     1    1     2   1         4.98
6     2    1     2   1         5.70
7     1    2     2   1         9.97
8     2    2     2   1         9.07
9     1    1     1   2         2.07
10    2    1     1   2         2.44
11    1    2     1   2         4.09
12    2    2     1   2         4.53
13    1    1     2   2         7.77
14    2    1     2   2         9.43
15    1    2     2   2        11.75
16    2    2     2   2        16.30
```

```
> rep(1:2,each=2,times=4)
```

Making patterned vectors is so common that there are additional arguments to rep to make these patterned vectors easy.

```
 [1] 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2
```

```
> out <- lm(advance.rate ~ (load+flow+speed+mud)^3,data=drill);anova(out)
```

Because we only have one replication, we can try to use the four-factor interaction as error. This doesn't work well, since it only leaves 1 df for error. Note that speed has an F ratio of about 143, but is only marginally "significant"!

```
Response: advance.rate
                Df  Sum Sq Mean Sq  F value  Pr(>F)
load             1   3.331   3.331   2.8821 0.33889
flow             1  43.494  43.494  37.6368 0.10287
speed            1 165.508 165.508 143.2197 0.05307 .
mud              1  20.885  20.885  18.0724 0.14708
load:flow        1   0.090   0.090   0.0779 0.82675
load:speed       1   1.416   1.416   1.2254 0.46770
load:mud         1   2.839   2.839   2.4569 0.36152
flow:speed       1   9.060   9.060   7.8400 0.21838
flow:mud         1   0.783   0.783   0.6778 0.56152
speed:mud        1  10.208  10.208   8.8333 0.20662
load:flow:speed  1   0.112   0.112   0.0971 0.80768
load:flow:mud    1   1.392   1.392   1.2049 0.47038
load:speed:mud   1   2.280   2.280   1.9730 0.39386
flow:speed:mud   1   0.130   0.130   0.1121 0.79428
Residuals        1   1.156   1.156
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

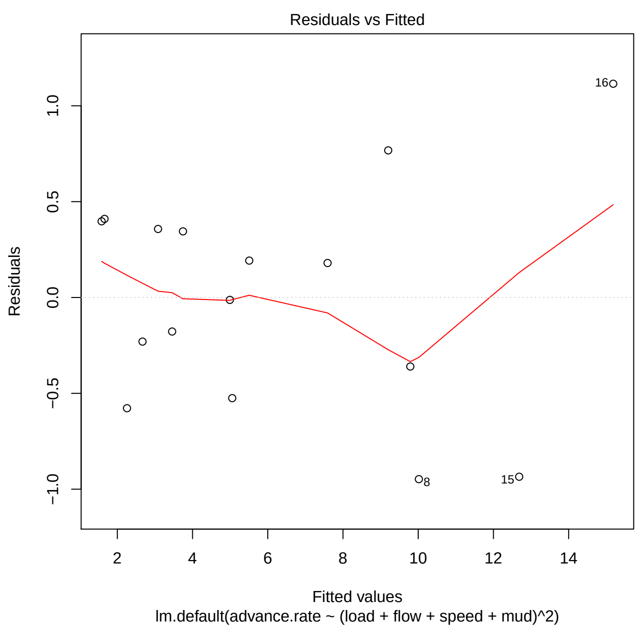
```
> out2 <- lm(advance.rate ~ (load+flow+speed+mud)^2,data=drill);anova(out2)
```

Let's try something more reasonable, where we only fit main effects and two factor interactions, pooling the three and four way interactions for error. The p-values we get are at best a guideline. It looks like flow, speed, mud, and some two factor interactions might be significant.

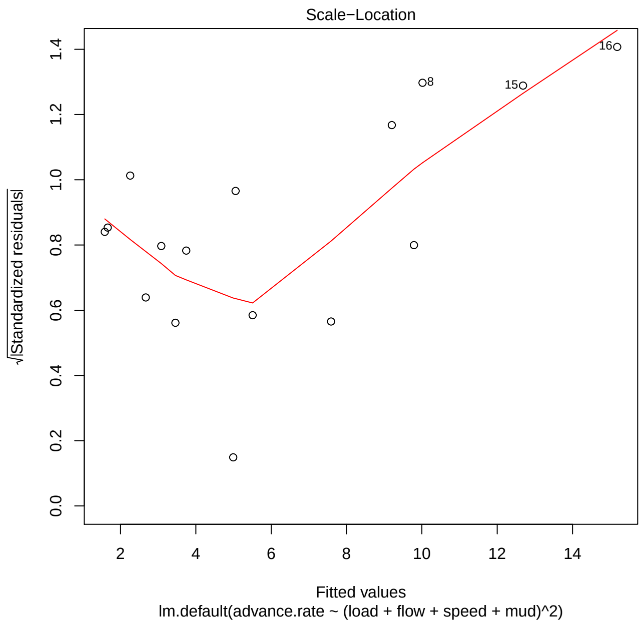
```
Analysis of Variance Table

Response: advance.rate
            Df  Sum Sq Mean Sq  F value     Pr(>F)
load         1   3.331   3.331   3.2847  0.129677
flow         1  43.494  43.494  42.8939  0.001243 **
speed        1 165.508 165.508 163.2247 5.227e-05 ***
mud          1  20.885  20.885  20.5968  0.006178 **
load:flow    1   0.090   0.090   0.0888  0.777744
load:speed   1   1.416   1.416   1.3966  0.290438
load:mud     1   2.839   2.839   2.8001  0.155118
flow:speed   1   9.060   9.060   8.9351  0.030477 *
flow:mud     1   0.783   0.783   0.7724  0.419694
speed:mud    1  10.208  10.208  10.0672  0.024735 *
Residuals    5   5.070   1.014
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
> plot(out2,which=1)
```

Both the residuals vs predicted plot here and the spread/level plot next indicate some increasing variance.



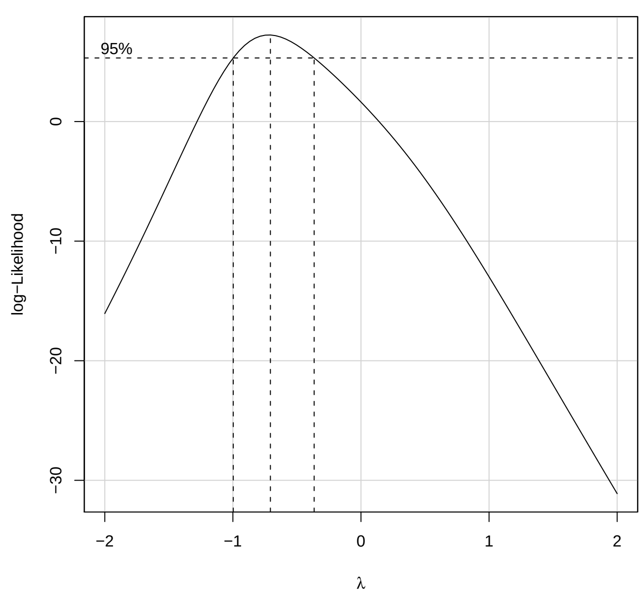
```
> plot(out2,which=3)
```




```
> boxCox(out2)
```

Box-Cox suggests the power -.75. Both -1 and -0.5 (barely) are within the confidence interval; for once, it's not the log! Note that the reciprocal is also easy to understand as a time to reach a drilling depth (rather than a drilling rate). We'll probably prefer the reciprocal because of this interpretation.

Also note, if we just had variables instead of variables in a data frame, then boxCox would have barfed on the name "load," because load is a function in R and boxCox was getting confused. We would need to rename it Load or something else.



```
> out2r <- lm(1/advance.rate~(load+flow+speed+mud)^2,data=drill)
```

Fit the second order model to reciprocal data.

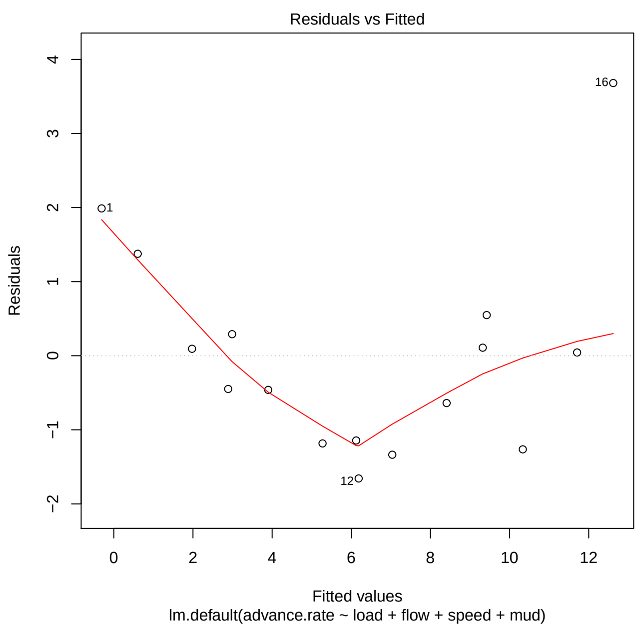
```
> out1 <- lm(advance.rate~load+flow+speed+mud,data=drill);anova(out1)
```

Well, before we declare world peace, let's start with a base model of just main effects. It's not that uncommon a place to start, and we might have gone there instead of the model with two factor interactions.

```
Analysis of Variance Table

Response: advance.rate
          Df  Sum Sq Mean Sq F value   Pr(>F)
load       1   3.331   3.331  1.2433 0.288598
flow       1  43.494  43.494 16.2365 0.001984 **
speed      1 165.508 165.508 61.7848 7.72e-06 ***
mud        1  20.885  20.885  7.7964 0.017517 *
Residuals 11  29.467   2.679
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

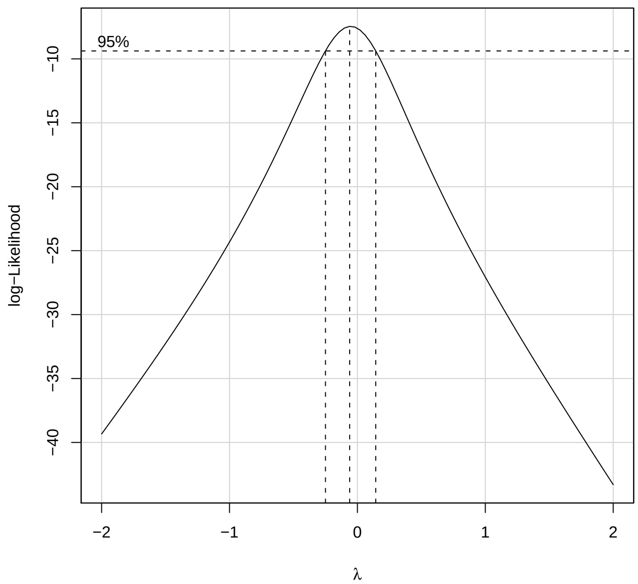
```
> plot(out1,which=1)
```

Oh my, oh my, oh my. This looks very bad. It should be flat, and it definitely is not. We have a flopping fish, and that usually means that you've either left out an important term or you're analyzing on the wrong scale.



```
> boxCox(out1)
```

What does BoxCox say? It is strongly recommending the log (you knew it had to come in, right?).
What is happening? Remember, interaction is scale dependent and can disappear. BoxCox is choosing log because things look better there, but a great deal of what is happening is that the interactions tend to disappear on the log scale. Thus log seems good when our base is that additive, main-effects model.




```
> #
```

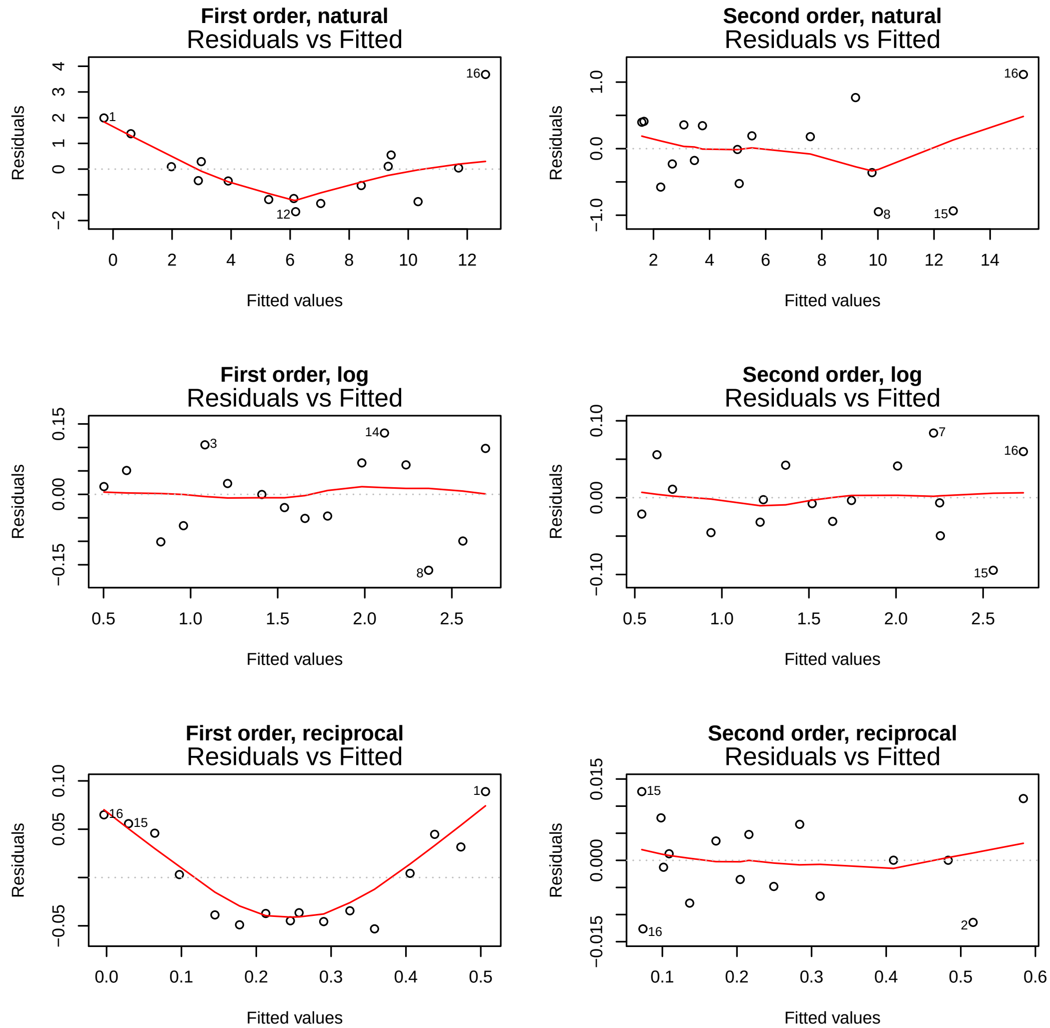
OK, let's be systematic and look at all the combinations of first and second order models together with natural, log or reciprocal responses.

```
> out1l <- lm(log(advance.rate)~load+flow+speed+mud,data=drill)
> out1r <- lm(1/advance.rate~load+flow+speed+mud,data=drill)
> out2l <- lm(log(advance.rate)~(load+flow+speed+mud)^2,data=drill)
```

```
> par(mfrow=c(3,2))
```

Make a 3x2 table of residual plots. The rows are natural, log, and reciprocal scales. The columns are first and second order models.

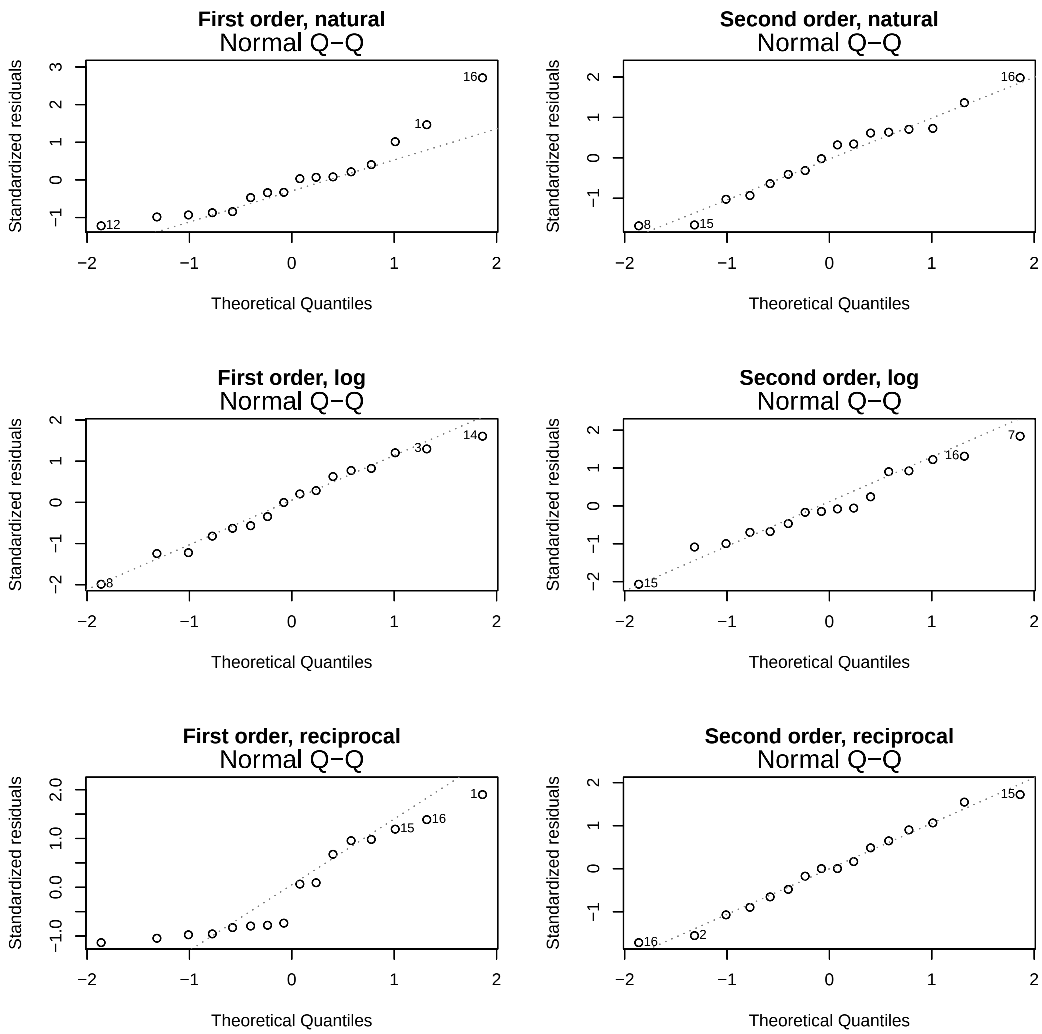
```
> plot(out1,which=1,main="First order, natural")
> plot(out2,which=1,main="Second order, natural")
> plot(out1l,which=1,main="First order, log")
> plot(out2l,which=1,main="Second order, log")
> plot(out1r,which=1,main="First order, reciprocal")
> plot(out2r,which=1,main="Second order, reciprocal")
```



```
> par(mfrow=c(3,2))
```

Make a 3x2 table of normal plots. The rows are natural, log, and reciprocal scales. The columns are first and second order models.

```
> plot(out1,which=2,main="First order, natural")
> plot(out2,which=2,main="Second order, natural")
> plot(out1l,which=2,main="First order, log")
> plot(out2l,which=2,main="Second order, log")
> plot(out1r,which=2,main="First order, reciprocal")
> plot(out2r,which=2,main="Second order, reciprocal")
```




```
> #
```

Overall, log works well on the first order model and surprisingly well on the second order model. The reciprocal works a little better than the log for the second order model, but there is not a lot of difference.
From a residuals point of view we have three reasonably passable models.

```
> anova(out1l)
```

On the log scale with the first order model, we have three highly significant factors and one slightly significant factor.

```
Analysis of Variance Table

Response: log(advance.rate)
          Df Sum Sq Mean Sq  F value    Pr(>F)
load       1 0.0676  0.0676   7.0171   0.02263 *
flow       1 1.3460  1.3460 139.7374 1.357e-07 ***
speed      1 5.3310  5.3310 553.4595 9.304e-11 ***
mud        1 0.4265  0.4265  44.2807 3.590e-05 ***
Residuals 11 0.1060  0.0096
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
> anova(out2l)
```

On the log scale with the second order model, we have the same basic conclusions with the notation of a marginally significant two factor interaction.

```
Analysis of Variance Table

Response: log(advance.rate)
           Df Sum Sq Mean Sq  F value    Pr(>F)
load        1 0.0676  0.0676  10.1188 0.0245108 *
flow        1 1.3460  1.3460 201.5036 3.124e-05 ***
speed       1 5.3310  5.3310 798.0978 1.041e-06 ***
mud         1 0.4265  0.4265  63.8536 0.0004956 ***
load:flow   1 0.0047  0.0047   0.7071 0.4387536
load:speed  1 0.0004  0.0004   0.0642 0.8101212
load:mud    1 0.0179  0.0179   2.6802 0.1625303
flow:speed  1 0.0101  0.0101   1.5093 0.2739060
flow:mud    1 0.0009  0.0009   0.1336 0.7296473
speed:mud   1 0.0385  0.0385   5.7677 0.0614982 .
Residuals   5 0.0334  0.0067
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
> anova(out2r)
```

On the reciprocal scale, we have all main effects highly significant and several significant interactions.

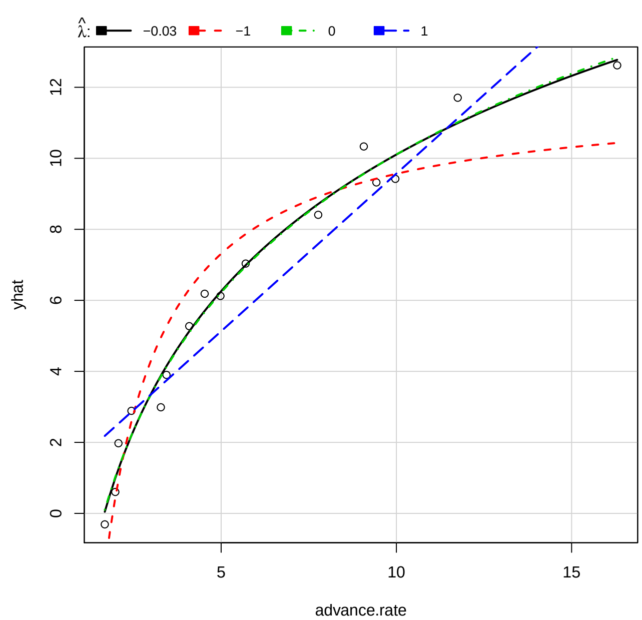
```
hi
Analysis of Variance Table

Response: 1/advance.rate
            Df   Sum Sq  Mean Sq   F value    Pr(>F)    
load         1 0.004330 0.004330   24.9820 0.0041111 ** 
flow         1 0.087967 0.087967  507.5816 3.202e-06 ***
speed        1 0.271974 0.271974 1569.3313 1.932e-07 ***
mud          1 0.018447 0.018447  106.4435 0.0001471 ***
load:flow    1 0.001595 0.001595    9.2012 0.0289701 *  
load:speed   1 0.001217 0.001217    7.0228 0.0454229 *  
load:mud     1 0.000035 0.000035    0.2015 0.6723157    
flow:speed   1 0.028765 0.028765  165.9812 5.018e-05 ***
flow:mud     1 0.001491 0.001491    8.6010 0.0325291 *  
speed:mud    1 0.001091 0.001091    6.2924 0.0539287 .  
Residuals    5 0.000867 0.000173                        
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
> #
```

Log was chosen to make the first order model look good, and only one of the two factor interactions is even close to being significant. Both models fit well, but I really like the simplicity we achieved on the log scale.

```
> inverseResponsePlot(out1)
```

If you are familiar with inverse response plots (you might have seen them in Stat 5302), here is one for the main effects model. The green curve corresponds to log, and this plot agrees that log is right.



```
> out2lb <- lm(log(advance.rate)~load*mud+speed*mud+flow,data=drill);anova(out2lb)
```

If you start with the log data and second order model and use our pooling rules, you can take out 4 terms and wind up with the following. There is a smaller error estimate with more degrees of freedom, which makes all terms more significant. What is really correct? Who knows. This is the price that you pay for having no estimate of pure error and needing to cobble it together from odds and ends.

```
Analysis of Variance Table

Response: log(advance.rate)
          Df Sum Sq Mean Sq  F value    Pr(>F)
load       1 0.0676  0.0676  12.2829  0.006674 **
mud        1 0.4265  0.4265  77.5103 1.022e-05 ***
speed      1 5.3310  5.3310 968.7920 1.788e-10 ***
flow       1 1.3460  1.3460 244.6004 7.845e-08 ***
load:mud   1 0.0179  0.0179   3.2534  0.104771
mud:speed  1 0.0385  0.0385   7.0013  0.026650 *
Residuals  9 0.0495  0.0055
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
> #
```

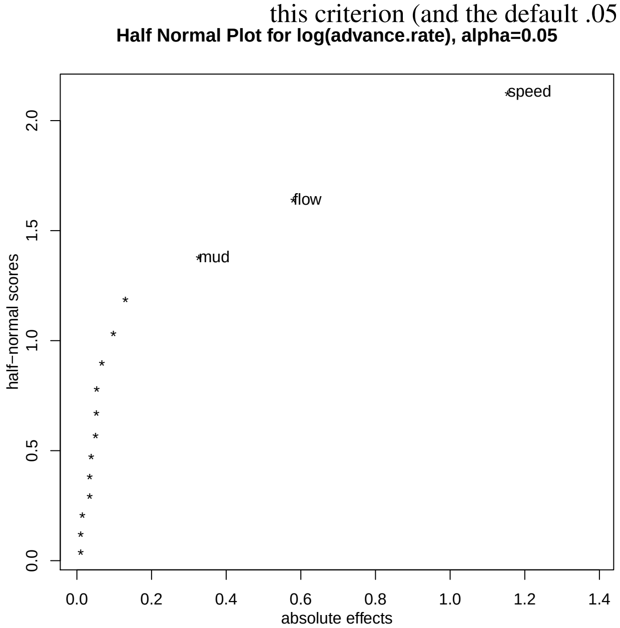
The FrF2 package, which Stat5303libs automatically loads, has a function to do Yates plots (which it calls Daniel plots, because Daniel invented it even if the plotted terms are Yates effects).

```
> outall <- lm(log(advance.rate)~load*flow*speed*mud,data=drill)
```

Make a model with all terms.

```
> DanielPlot(outall,half=TRUE,code=FALSE)
```

Make the plot. There are many options. I like the half normal plot. Coding replaces the factor names by A, B, C, etc, which can be helpful for interaction labeling. The function automatically marks the significant effects according to Lenth's pseudo standard error. Using this criterion (and the default .05 level), only flow, speed, and mud are significant.

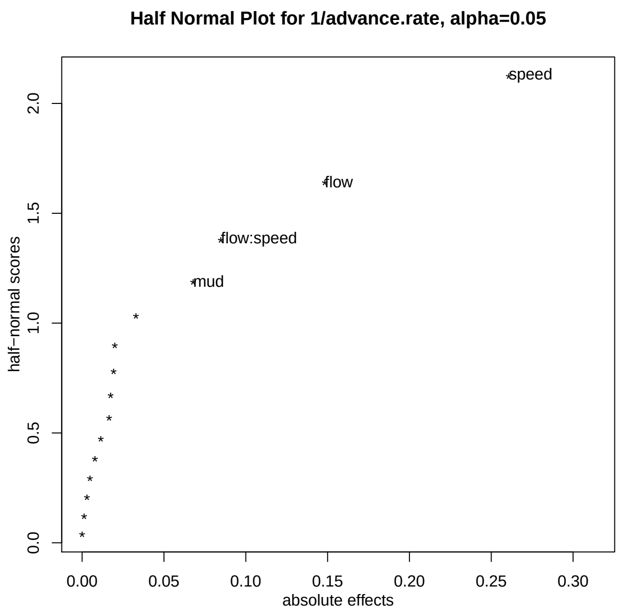


```
> outallr <- lm(1/advance.rate ~ load*flow*speed*mud,data=drill)
```

Redo for reciprocal scale.

```
> DanielPlot(outallr,half=TRUE)
```

If you look back at the anova for the 2fi model on the reciprocal scale, a lot of things looked "significant." Here, only four do. So what happened? The Lenth approach assumes model sparsity. That is, it assumes that most effects are null. When lots of effects start looking nonnull, the Lenth technique mixes some of them in when it computes its pseudo standard error. That PSE gets bigger, so fewer things look big relative to the PSE.




```
> y<-c(8,4,53,43,31,9,12,36,79,68,73,8,77,38,49,23)/100
```

Data from Davies (1954) via Box and Meyer (1986). Response is the yield of isatin under different production conditions. Factors are: A – acid strength; B – time; C – amount of acid; D – temperature. These data are also in standard order.

```
> A <- factor(rep(1:2,times=8))

> B <- factor(rep(1:2,each=2,times=4))

> C <- factor(rep(1:2,each=4,times=2))

> D <- factor(rep(1:2,each=8))
```

```
> out <- lm(y~A+B+C+D);anova(out)
```

Maybe D is significant. Alternatively, *everything* is significant, but we can't see it because we are using a large interaction for error, making other significant effects look insignificant. Without an estimate of pure error, we cannot tell which situation we are in.

```
Analysis of Variance Table

Response: y
          Df  Sum Sq Mean Sq F value Pr(>F)
A          1 0.14631 0.14631  2.8052 0.1221
B          1 0.00181 0.00181  0.0346 0.8558
C          1 0.02326 0.02326  0.4459 0.5181
D          1 0.29976 0.29976  5.7473 0.0354 *
Residuals 11 0.57372 0.05216
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
> outall <- lm(y~A*B*C*D)
```

Get ready for Daniel plot.

```
> DanielPlot(outall)
```

Looks like a whole lot of nothing. Nothing is significant. What happened to D? It's one of the foibles of factorial anova modeling. Many people (including me sometimes) neglect the multiple testing aspect of factorial anova. We may be doing a lot of tests, and some of them are bound to look significant just by chance. In this case, D may be the biggest effect, but it's not unusually big for the largest of 15.