Notes for Thursday February 12, 1998

Today we consider two very common designs:

- Latin Square designs let you account for two blocking factors, instead of the one factor accounted for by randomized block designs;
- Split-plot designs are one way to handle "nesting" in the responses (for example, pupils nested in a classroom).

1 A Latin Square Design

We have seen, with randomized block designs, that including the block in the model can eliminate its effect on the outcome, and make tests of the treatments more powerful. Carefully balanced designs can eliminate more than one blocking variable in an experiment or study. When the number of treatments equals the number of blocks, the most common design is *Latin Squares*.

Suppose that 4 cars and four drivers are employed in a study of possible differences between four gasoline additives. Even though the cars are identical models, slight systematic differences are likely to occur in their performance, and even though each driver may do her best to drive the car using the experimental protocol, slight systematic differences can occur from driver to driver. It would be desirable to design (and analyze) an experiment that eliminates both the driver-to-drive and automobile-to-automobile effects. The Latin Square arrangement below allows this to be done.

		cars			
		1	2	3	4
	I	A	B	D	C
drivers	II	D	C	A	B
	III	B	D	C	A
	IV	C	A	B	D

with cars (I,II,III, and IV), drivers (1, 2, 3, and 4) and additives [treatments] (A,B,C,and D).

Data will be assigned to each combination, so that data will be indexed by three things (car, driver, and additive), each with 4 levels, but there will only be 16 data points. We do **not** have a complete replication of the experiment.

The model might be written

$$Y_{ijk} = \mu + d_i + c_j + a_k + \epsilon_{ijk}$$

There is not much hope of estimating interactions.

Here is some data from this experiment.

		cars			
		1	2	3	4
	I	A 21	B 26	D 20	C 25
drivers	II	D 23	C 26	A 20	B 27
	III	B 15	D 13	C 16	A 16
	IV	C 17	A 15	B 20	D 20

Lets do a quick analysis of these data, first entering them, and then an ANOVA and plots. Here is `gas.dat`:

```
A 21
B 26
D 20
C 25
D 23
C 26
A 20
B 27
B 15
D 13
C 16
A 16
C 17
A 15
B 20
D 20
```

and here is the data being entered.

```
402 > delme <-read.table("gas.dat")
402 > delme
   V1 V2
 1  A 21
 2  B 26
 3  D 20
 4  C 25
 5  D 23
 6  C 26
 7  A 20
 8  B 27
 9  B 15
10  D 13
11  C 16
12  A 16
13  C 17
14  A 15
15  B 20
16  D 20
402 > gas <- delme$V1
402 > resp <- delme$V2
402 > car <- rep(1:4, len=16)
402 > car
 [1] 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4
402 > driver <- rep(1:4,rep(4,4))
```

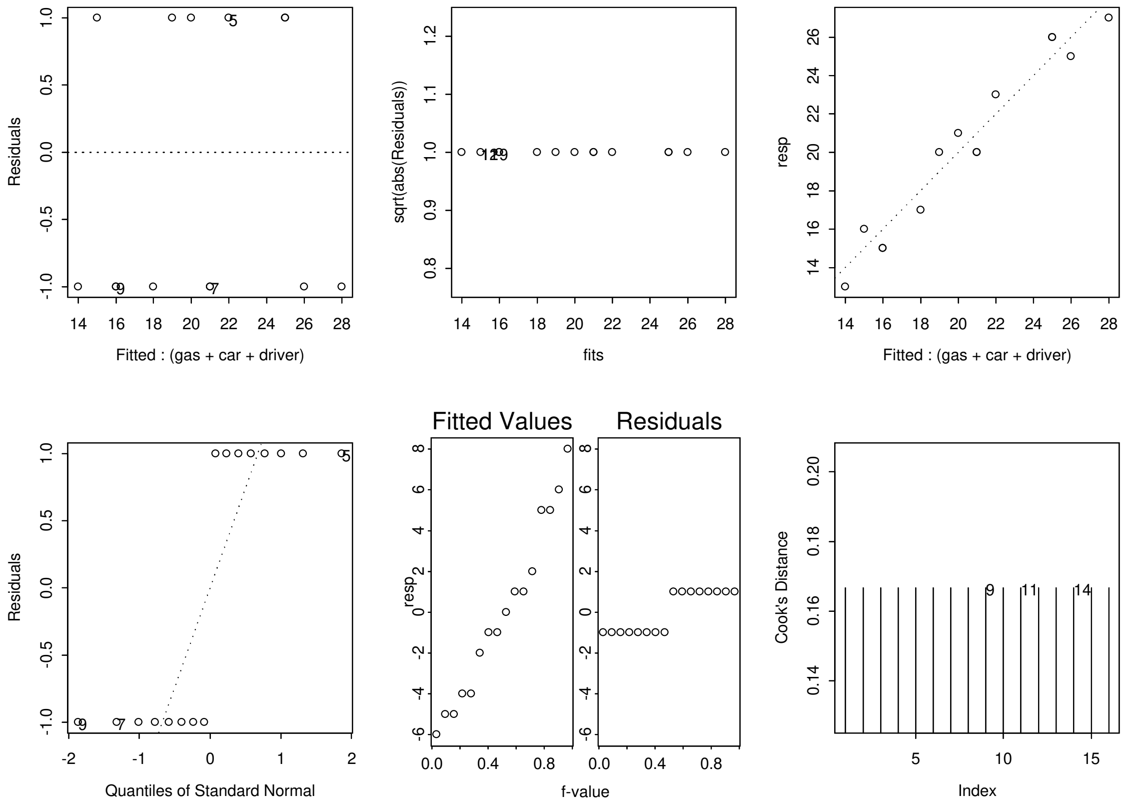
```
402 > driver
 [1] 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4
402 > gas.frame <- data.frame(resp, gas, car=as.factor(car), driver=as.factor(driver) )
```

So here is my constructed data set

```
402 > gas.frame
   resp gas car driver
 1   21   A   1      1
 2   26   B   2      1
 3   20   D   3      1
 4   25   C   4      1
 5   23   D   1      2
 6   26   C   2      2
 7   20   A   3      2
 8   27   B   4      2
 9   15   B   1      3
10   13   D   2      3
11   16   C   3      3
12   16   A   4      3
13   17   C   1      4
14   15   A   2      4
15   20   B   3      4
16   20   D   4      4
```

Now onto an analysis.

```
402 > anova(mymod <- aov(resp ~ gas + car + driver,data=gas.frame) )
Analysis of Variance Table

Response: resp

Terms added sequentially (first to last)
          Df Sum of Sq  Mean Sq F Value     Pr(F)
gas        3        40 13.33333       5 0.0451975
car        3        24  8.00000       3 0.1169598
driver     3       216 72.00000      27 0.0006987
Residuals  6        16  2.66667
```



Wow, look at those residuals!!! What's going on???

2 Split-Plot Designs

A split-plot design contains more than one source of error. This can arise because factors are applied at different scales, as in the guayule example below.

Split-plots are also encountered because of restrictions on the randomization. For example, an experiment involving oven temperature and baking time will probably not randomize the oven temperature totally, but rather only change the temperature after all of the runs for that temperature have been made. This type of design is often mistakenly analyzed as if there were no restrictions on the randomization (an indication of this can be p-values that are close to 1). See Hicks (1973) and Daniel (1976). S-PLUS includes the guayule data frame which is also discussed in Chambers and Hastie (1992). This experiment was on eight varieties of guayule (a rubber producing shrub) and four treatments on the seeds. Since a flat (a shallow box for starting seedlings) was not large enough to contain all 32 combinations of variety and treatment, the design was to use only a single variety in each flat and to apply each treatment within each flat. Thus the flats each consist of four sub-plots. This is a split-plot design since flats are the experimental unit for varieties, but the sub-plots are the experimental unit for the treatments. The response is the number of plants that germinated in each sub-plot.

The guayule data frame, a design object, has 96 rows and 5 columns. The guayule is a Mexican plant from which rubber is manufactured. Batches of 100 seeds of eight varieties (variety) of guayule were given one of four treatments (treatment), and planted; the number of plants that came up in each batch (plants) was recorded. In order to separate the effect of which flat (planting tray) a batch was planted on from the effect of the treatment, each flat was planted with seeds from only one variety, but all the treatments were represented in each flat.

This data frame contains the following columns:

`variety`	factor with levels V1 through V8 labeling the variety of guayule.
`treatment`	factor with levels T1 through T4 labeling the treatment given to the seeds.
`reps`	1, 2, or 3. All eight varieties of plant are represented once in each rep.
`plants`	numeric vector givng the number seeds out of a batch of 100 that germinated.
`flats`	factor labeling in which of 24 flats (3 of each variety) a batch was planted.

To analyze a split-plot design like this, put the variable that corresponds to the whole plot in an Error term in the formula of the aov call:

```
402 > gua.aov1 <- aov(plants ~ variety*treatment + Error(flats), guayule)
```

As usual, you can get an ANOVA table with summary:

```
402 > summary(gua.aov1)
Error: flats
          Df Sum of Sq  Mean Sq  F Value     Pr(F)
  variety  7   763.156 109.0223 1.232036 0.3420697
Residuals 16  1415.833  88.4896

Error: Within
                  Df Sum of Sq  Mean Sq  F Value        Pr(F)
        treatment  3  30774.28 10258.09 423.4386 0.00000e+000
variety:treatment 21   2620.14   124.77   5.1502 1.32674e-006
        Residuals 48   1162.83    24.23
```

This shows varieties tested with the error from flats, while treatment and its interaction with variety are tested with the within-flat error (which is substantially smaller).

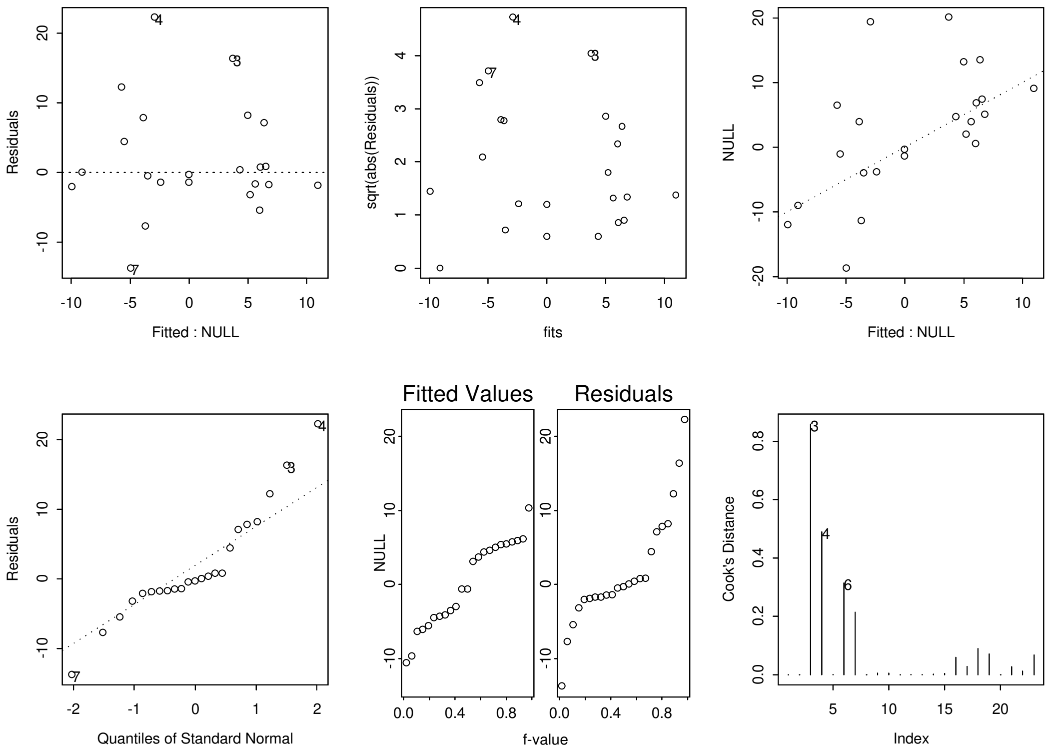
The guayule data actually represent an experiment in which the flats were grouped into replicates—making three sources of error, or a split-split-plot design. To model this we put more than one term inside the Error term:

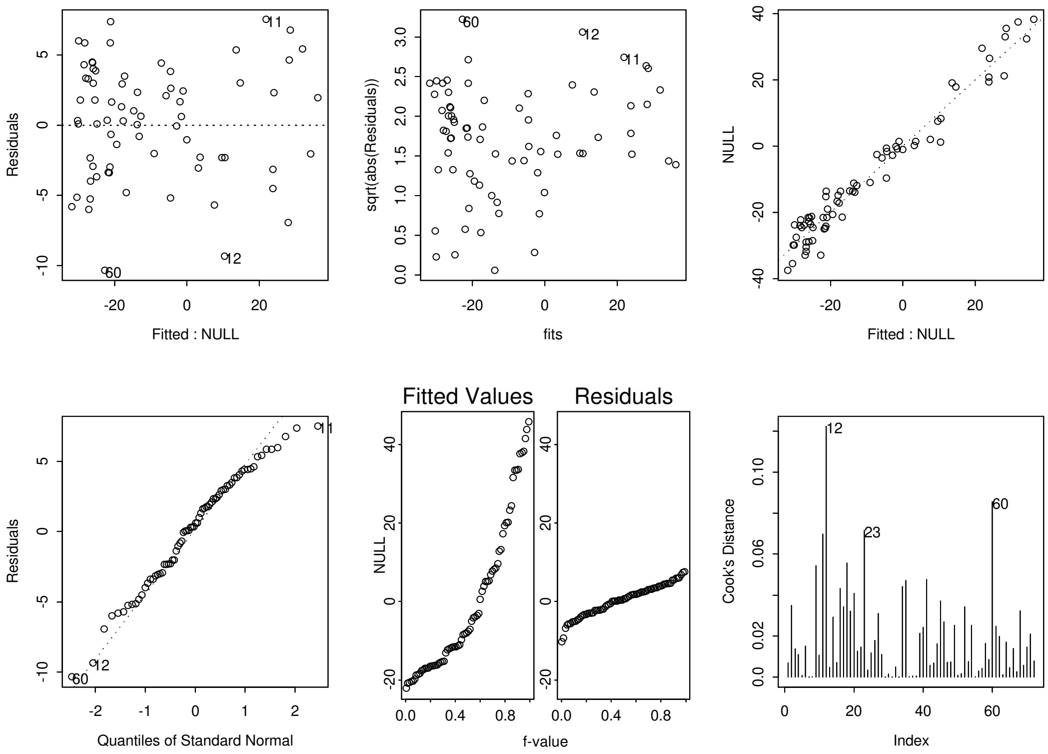
```
> gua.aov2 <- aov(plants ~ variety*treatment + Error(reps/flats), guayule)
> summary(gau.aov2)
Error: reps
          Df Sum of Sq  Mean Sq F Value Pr(F)
Residuals  2  38.58333 19.29167
```

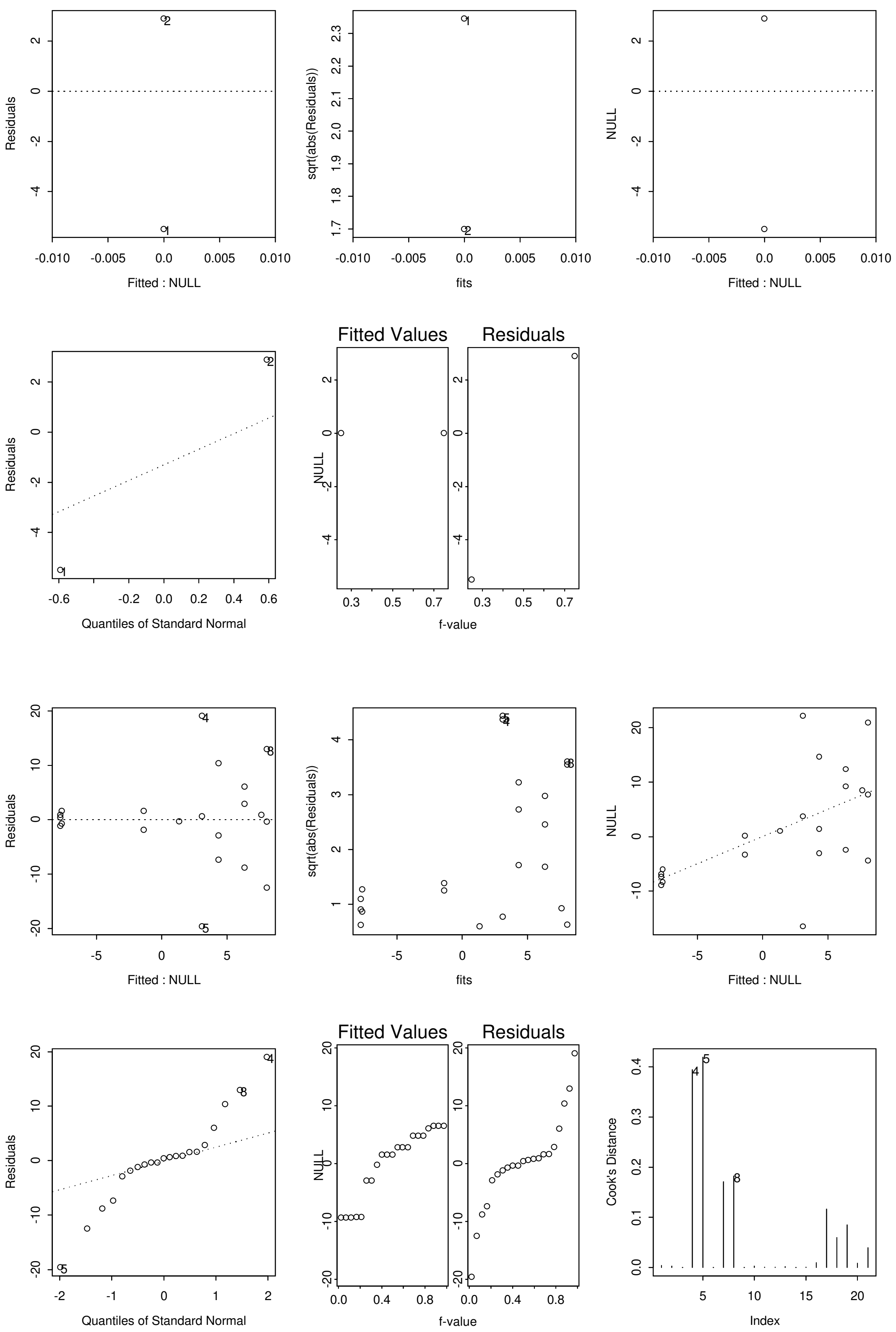
```
Error: flats %in% reps
         Df Sum of Sq  Mean Sq  F Value     Pr(F)
  variety  7   763.156 109.0223 1.108232 0.4099625
Residuals 14  1377.250  98.3750

Error: Within
                  Df Sum of Sq  Mean Sq  F Value        Pr(F)
        treatment  3  30774.28 10258.09 423.4386 0.00000e+000
variety:treatment 21   2620.14   124.77   5.1502 1.32674e-006
        Residuals 48   1162.83    24.23
```

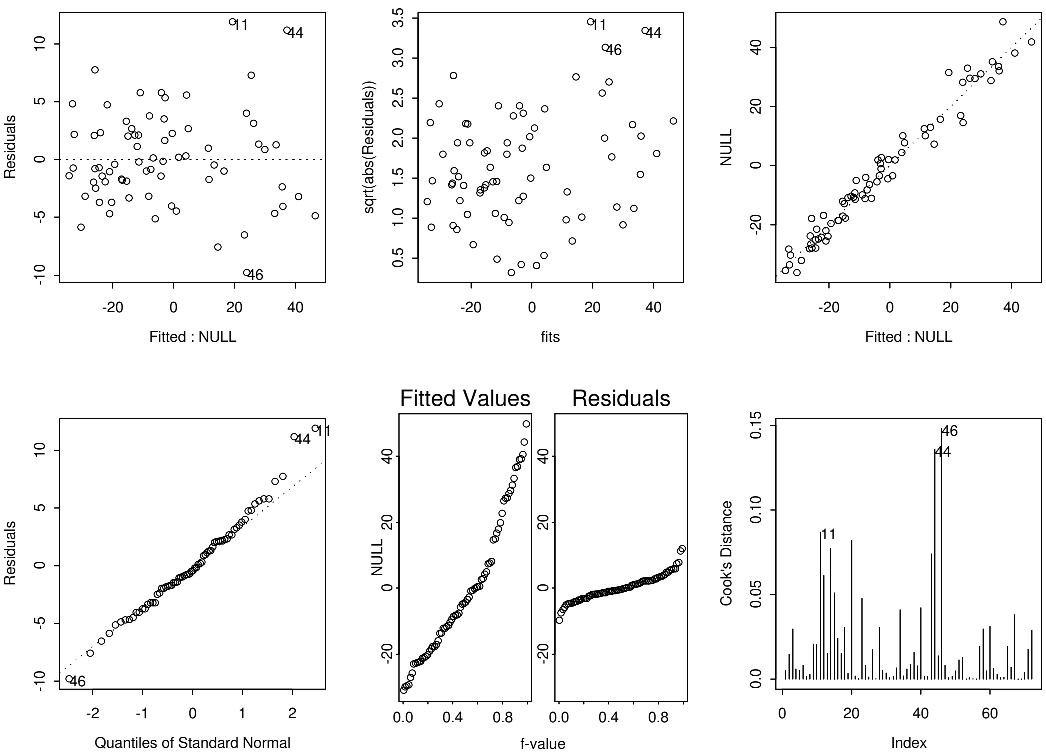
The Error term could also have been specified as `Error(reps + flats)`. However, the specification `Error(flats + reps)` would not give the desired result (the sequence within the Error term is significant); explicitly stating the nesting is preferred. Note that only one Error term is allowed.

```
402 > par(mfrow=c(2,3))
402 > gua.fm1 _ update(gua.aov1,qr=T)
402 > gua.fm2 _ update(gua.aov2,qr=T)
402 > names(gua.fm1)
[1] "(Intercept)" "flats"       "Within"
402 > plot(gua.fm1$flats,title="Guayule Error(flats) flats")
402 > plot(gua.fm1$Within,title="Guayule Error(flats) within")
402 > names(gua.fm2)
[1] "(Intercept)"     "reps"            "flats %in% reps" "Within"
402 > plot(gua.fm2$reps,title="Guayule Error(reps/flats) reps")
402 > plot(gua.fm2$"flats %in% reps",title="Guayule Error(reps/flats) flats")
402 > plot(gua.fm2$Within,title="Guayule Error(reps/flats) Within")
```


Residuals
Fitted : NULL
sqrt(abs(Residuals))
fits
NULL
Fitted : NULL
Residuals
Quantiles of Standard Normal
Fitted Values
Residuals
NULL
f-value
Cook's Distance
Index


Residuals
Fitted : NULL
sqrt(abs(Residuals))
fits
NULL
Fitted : NULL
Residuals
Quantiles of Standard Normal
Fitted Values
Residuals
NULL
f-value
Cook's Distance
Index


Residuals
Fitted : NULL
sqrt(abs(Residuals))
fits
NULL
Fitted : NULL
Residuals
Quantiles of Standard Normal
Fitted Values
Residuals
NULL
f-value
Residuals
Fitted : NULL
sqrt(abs(Residuals))
fits
NULL
Fitted : NULL
Residuals
Quantiles of Standard Normal
Fitted Values
Residuals
NULL
f-value
Cook's Distance
Index


Residuals
Fitted : NULL
sqrt(abs(Residuals))
fits
NULL
Fitted : NULL
Residuals
Quantiles of Standard Normal
Fitted Values
Residuals
NULL
f-value
Cook's Distance
Index
11
44
46