Notes for Thursday February 5, 1998

1 More on Interactions

1.1 Notation for Effects and Interactions

Since we've been talking about effects and interactions in SPLUS it would be a good idea to summarize SPLUS's notation for them.

There are many different notations for main effects and interactions. Many books just use model formulae and subscripts. The S language, and most other computer packages, use some variant of what has come to be known as GLIM notation. (GLIM was an early statistical package for Generalized Linear Interactive Modelling. GLIM is still available, but largely superseded by programs like S). The general idea is that there is a model "calculus" for the formulae. We have already seen some examples.

Consider Y a response vector and R is a factor with n_R levels, (Remember that a factor is just a variable whose entries are levels rather than numbers), C is a factor with n_C levels and L is a factor with n_L levels. We'll call them *rows* (R), *columns* (C) and *layers* (L), but they could be any "factor" (like dose of a drug, male/female, type of car, etc.). Here are some examples:

- The formula $Y \sim R + C$ specifies an *additive* model, that is, the response can be expressed as a linear combination of the row levels and the column level, with no interactions. In math notation we could write this as

  $$Y_{ijk} = \mu + \alpha_i + \beta_j + \epsilon_{ijk} \quad \text{or} \quad Y_{ijk} = \mu + R_i + L_j + \epsilon_{ijk}$$

  (just replacing Greek α's amd β's with the names of the factors, for clarity). i and j index row and column levels, and k indexes replications within cells.

- The formula $Y \sim R + C + L$ says that the response can be expressed as a linear combination of row levels, column levels, and layer levels. In math notation, this translates into

  $$Y_{ijkl} = \mu + R_i + C_j + L_k + \epsilon_{ijkl}$$

  (i, j, k index levels of the factors, and l indexes replication within cells). The expected response lives in the linear space of linear combinations of row, column and layer effects.

- The formula $Y \sim R + C + L + C : L$ says that we have main effects for rows, columns, and layers, but also allow for a column times layer interaction. In math that is,

  $$Y_{ijkl} = \mu + R_i + C_j + L_k + CL_{jk} + \epsilon_{ijkl}$$

- A shorthand form for the above model is $Y \sim R + C * L$ where the $C * L$ explicitly expands into $C + L + C : L$

- One could write the model $Y \sim R + C : L$ where the $C : L$ is meant to refer to the interaction without the corresponding main effects, but this rarely makes sense (and S won't easily allow it).

- A final example is $Y \sim R * C * L$ which expands out to $Y \sim R + C + L + R : C + R : L + C : L + R : C : L$, and in math notation,

$$Y_{ijkl} = \mu + R_i + C_j + L_k + CL_{jk} + RC_{ij} + RL_{ik} + RCL_{ijk} + \epsilon_{ijkl}$$

1.2 Fooling Around with Interactions

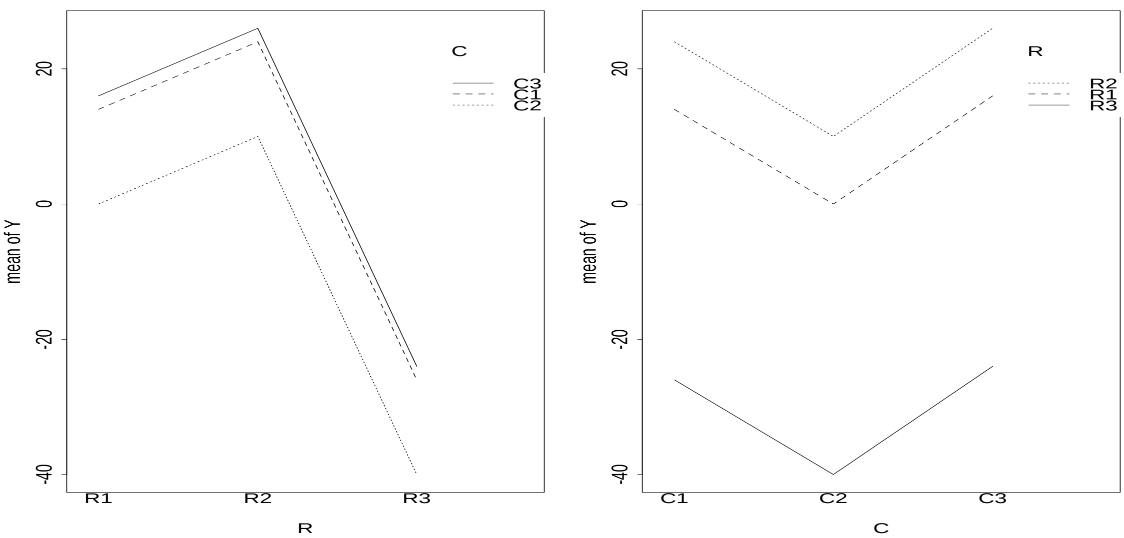
The concept of interaction is both simple and tricky at the same time. Consider a two-way layout of data, indexed by rows i and columns j. Let's call the data Y_{ijk}. An **additive** model for the data is

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \epsilon_{ijk}$$

where the errors have some (usually normal) distribution with mean zero and constant variance. Let's look at a table of cell means that satisfies this model, with $\mu = 0$.

14	0	16	10
24	10	26	20
-26	-40	-24	-30
4	-10	6	0

This data is perfectly additive—the entries in the table are exactly the row effect plus the column effect. An interaction plot for the data looks like this:



Interaction is the term that describes the departure from the additive model—the extent to which the traces in an interaction plot are not parallel.

Our discussion of interactions last time was based on "unstructured" interactions—that is, we assume there are an additional $I \cdot J$ parameters (subject to sum-to-zero constraints) that describe the interaction (departure from the additive model) in *every* cell of the table.

- When there is more than one observation per cell we can estimate *both* all of the interaction terms (how does each cell mean differ from the pure additive model) *and* the pure error (what is the variability of the observations around the cell mean). We did this with the **used car offer** and **human growth hormone** examples from last time. I.e. we were looking at the model

$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

subject to sum-to-zero constraints.

- When we only have one observation per cell, there is not enough data to estimate all $(I-1) \times (J-1)$ extra parameters. Therefore, we usually assign interaction to the error (ϵ_{ijk} above); this is equivalent to adding the linear constraints that all $(\alpha\beta)_{ij} = 0$. We did this with the **penicillin** example last time; i.e. we are fitting exactly the model

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \epsilon_{ijk}$$

Even with only one observation per cell, we might sometimes choose to dissect the error into an interaction component and an error component. This is most commonly done when there is a specific type of interaction that we are looking for. This gives us additional constraints, so we don't actually have to estimate all $(I-1) \times (J-1)$ extra parameters.

For example, consider the following made-up data.

```
402 > mydata
   V1 V2 V3
1  18  A  a
2  -6  A  b
3  20  A  c
4  29  B  a
5  16  B  b
6  13  B  c
7 -28  C  a
8 -36  C  b
9 -17  C  c
402 > attach(mydata)  # so I don't always have to say ``data=mydata'' in aov()
402 > mymod <- aov(V1 ~ V2 + V3 )
```

So we have some numbers in a 3×3 layout, and we fit the simple model with row and column effects. The ANOVA table is

```
Analysis of Variance Table

Response: V1

Terms added sequentially (first to last)
          Df Sum of Sq  Mean Sq  F Value     Pr(F)
V2         2  3640.667 1820.333 22.51959 0.0066532
V3         2   422.000  211.000  2.61031 0.1881914
Residuals  4   323.333   80.833
```

Plowing ahead, the researcher believes there is cause to worry about a linear times linear interaction—an effect which is linear in rows and columns. To test this theory she creates a one degree of freedom) "effect", i.e. another predictor in the linear regression version of ANOVA, to add to her model. The particular effect is

```
402 > as.numeric(V2)
[1] 1 1 1 2 2 2 3 3 3
402 > as.numeric(V3)
[1] 1 2 3 1 2 3 1 2 3
402 > LinXlin <- as.numeric(V2) * as.numeric(V3)
402 > LinXlin
[1] 1 2 3 2 4 6 3 6 9
```

which is easily seen to be linear in both components. The resulting ANOVA and summary is

```
402 > mymod <- aov(V1 ~ V2 + V3 + LinXlin)
402 > anova(mymod)
Analysis of Variance Table

Response: V1

Terms added sequentially (first to last)
          Df Sum of Sq  Mean Sq  F Value     Pr(F)
V2         2  3640.667 1820.333 18.01815 0.0213049
V3         2   422.000  211.000  2.08853 0.2702472
LinXlin    1    20.250   20.250  0.20044 0.6847201
Residuals  3   303.083  101.028
```

Notice, all we did was add the new effect to the model and the SS for that effect was extracted from the SS for residuals and assigned to the named effect. In these data there doesn't seem to be much of a linear times linear effect. There are two ways to write the model we fit. Most explicitly

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \tau \cdot i \cdot j + \epsilon_{ijk}$$

or to be more suggestive,

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \tau \cdot LL_k + \epsilon_{ijk}$$

where LL_k is the linear times linear vector: $LL_k = i \cdot j$ if the k^{th} observation is in the ij^{th} cell. Comparing with the unrestricted interaction ANOVA model

$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

we see that the constraint we have made on the interaction terms is that $(\alpha\beta)_{ij} = \tau \cdot LL_k = \tau \cdot i \cdot j$, i.e. we have reduced the model from having $(I-1)\cdot(J-1)$ interaction parameters $(\alpha\beta)_{ij}$ to having just one interaction parameter τ.

How else might we break apart the SS for residuals? In the first analysis there were 4 degrees of freedom for error. There are **MANY** ways we can partition those 4 degrees of freedom, but a common method is into linear and quadratic components.

Below, we create linear and quadratic components of the interaction.

```
402 > LL <- as.numeric(V2) * as.numeric(V3)          # LL = i   * j
402 > LQ <- as.numeric(V2) * (as.numeric(V3))^2      # LQ = i   * j^2
402 > QQ <- (as.numeric(V2))^2 * (as.numeric(V3))^2  # QQ = i^2 * j^2
402 > QL <- (as.numeric(V2))^2 * (as.numeric(V3))    # QL = i^2 * j
402 > data.frame(LL,LQ,QL,QQ)
  LL LQ QL QQ
1  1  1  1  1
2  2  4  2  4
3  3  9  3  9
4  2  2  4  4
5  4  8  8 16
6  6 18 12 36
7  3  3  9  9
8  6 12 18 36
9  9 27 27 81
```

Since there were 4 df for error in the original additive model, if we now do an ANOVA with all these effects we should find ourselves with no degrees of freedom for error, as we do.

```
402 > mymod <- aov(V1 ~ V2 + V3 + LL + LQ + QL + QQ)
402 > anova(mymod)
Analysis of Variance Table

Terms added sequentially (first to last)
   Df Sum of Sq  Mean Sq
V2  2  3640.667 1820.333
V3  2   422.000  211.000
LL  1    20.250   20.250
LQ  1    44.083   44.083
QL  1   168.750  168.750
QQ  1    90.250   90.250
```

Just as a check, lets fit the model with no QQ term (that is the model that allows for all interactions except the quadratic by quadratic one). It looks like

```
402 > anova(mymod <- aov(V1 ~ V2 + V3 + LL + LQ + QL))
Analysis of Variance Table

Terms added sequentially (first to last)
          Df Sum of Sq  Mean Sq  F Value     Pr(F)
V2         2  3640.667 1820.333 20.16990 0.1555306
V3         2   422.000  211.000  2.33795 0.4197422
LL         1    20.250   20.250  0.22438 0.7183758
LQ         1    44.083   44.083  0.48846 0.6116710
QL         1   168.750  168.750  1.86981 0.4019821
Residuals  1    90.250   90.250
```

2 2^k Factorial Designs—Examples

Factorial designs with each factor at two levels are extremely important in practice. Many experiments, especially industrial ones, have factors at 2 levels. Things like (off,on), (high, low), (treatment, placebo).

Factorial designs, and in particular 2^k factorial design are important because:

- They require relatively few runs per factor studied.
- It is possible to run *fractional* designs that are even more economical—at some expense in accuracy/estimability.
- They are easy to interpret, and some simple effects can be seen with common sense and a little arithmetic. (Hence, it is easier to convince your boss of what is happening)

There are many standard notations for a (say) 2^3 design. If the three factors are T, C, and K, each at 2 levels, then here are just some of the ways such a design is represented.

Run	T	C	K		T	C	K
1	−	−	−	1	0	0	0
2	+	−	−	t	1	0	0
3	−	+	−	c	0	1	0
4	+	+	−	tc	1	1	0
5	−	−	+	k	0	0	1
6	+	−	+	tk	1	0	1
7	−	+	+	ck	0	1	1
8	+	+	+	tck	1	1	1

Here is data from a particular experiment, coded in a couple of different ways. The original data were part of a pilot plant investigation of a process and have been simplified for this example. There are two quantitative predictors (temperature and concentration) and a qualitative predictor (catalyst). The response is the chemical yield.

test condition number	temp (C) T	concentration (%) C	catalyst (A or B) K	yield (grams) y
		Original units of variables		
1	160	20	A	60
2	180	20	A	72
3	160	40	A	54
4	180	40	A	68
5	160	20	B	52
6	180	20	B	83
7	160	40	B	45
8	180	40	B	80
		Coded units of variables		
1	−	−	−	60
2	+	−	−	72
3	−	+	−	54
4	+	+	−	68
5	−	−	+	52
6	+	−	+	83
7	−	+	+	45
8	+	+	+	80

with

Temperature		Concentration		Catalyst	
−	+	−	+	−	+
160	180	20	40	A	B

Lets try a quick analysis of these data, in the first form. Here is the data

```
Run Temp  Conc Caty Yield
1   160   20   A    60
2   180   20   A    72
3   160   40   A    54
4   180   40   A    68
5   160   20   B    52
6   180   20   B    83
7   160   40   B    45
8   180   40   B    80
```

and reading the data into S

```
402 > chem <- read.table("chemical.dat", header=T)
```

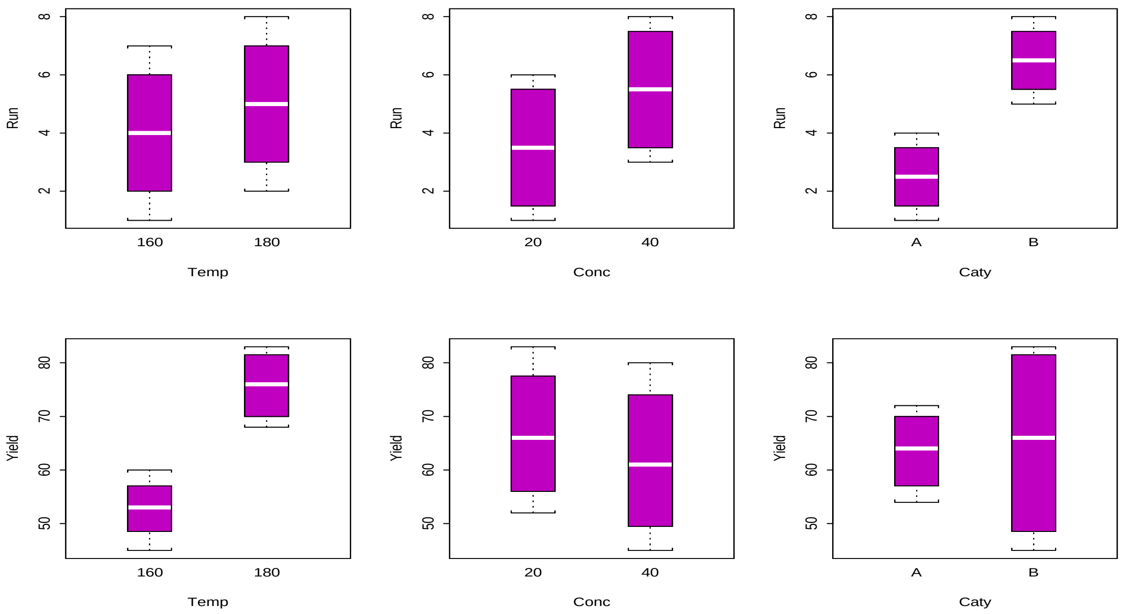
```
402 > chem
  Run Temp Conc Caty Yield
1   1  160   20    A    60
2   2  180   20    A    72
3   3  160   40    A    54
4   4  180   40    A    68
5   5  160   20    B    52
6   6  180   20    B    83
7   7  160   40    B    45
8   8  180   40    B    80
```

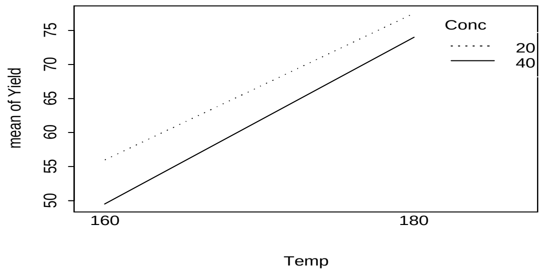
SPLUS will have no trouble interpreting the character values of `Caty` as factor levels, but unless we explicitly tell it to treat the numerical data in `Temp` and `Conc` as factor levels, it will treat them as continuous predictors (like regression, instead of like ANOVA). Here's a simple fix (there are many others possible):

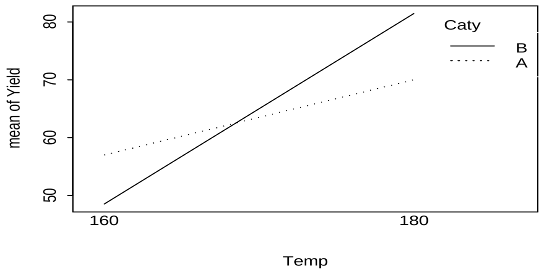
```
402 > chem$Temp _ as.factor(chem$Temp)
402 > chem$Conc _ as.factor(chem$Conc)
```

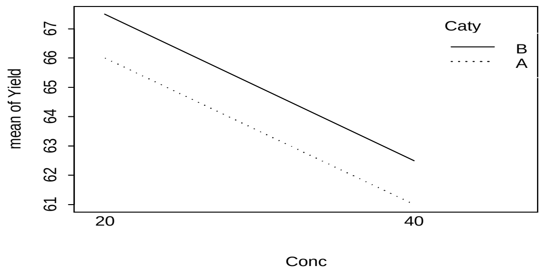
Now we look at the standard plots

```
402 > par(mfrow=c(2,3))
402 > plot.factor(chem)
402 > attach(chem)
402 > par(mfrow=c(2,2))
402 > interaction.plot(Temp,Conc,Yield)
402 > interaction.plot(Temp,Caty,Yield)
402 > interaction.plot(Conc,Caty,Yield)
402 > detach(chem)
```



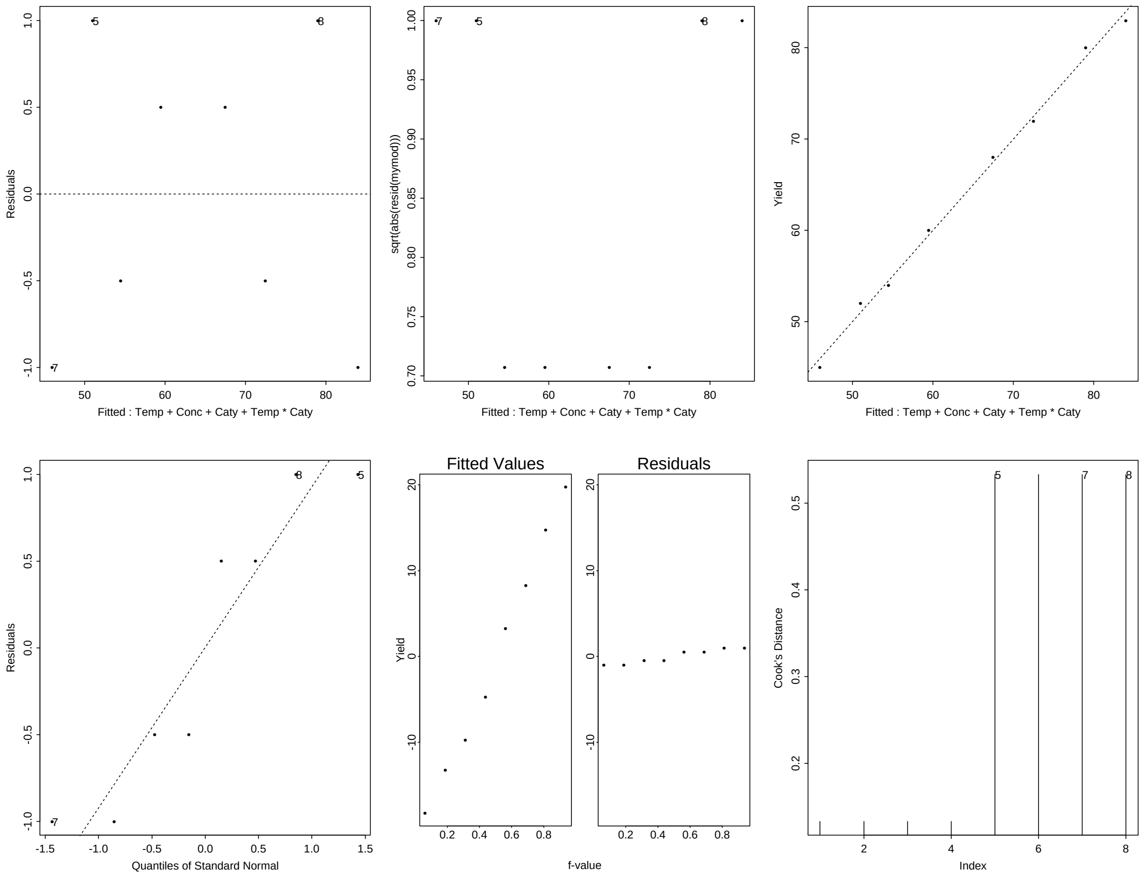






It appears that **Temp:Caty** is the only interesting interaction, so that is the one we will fit.

```
402 > mymod <- aov(Yield ~ Temp + Conc + Caty + Temp:Caty, data=chem)
402 > anova(mymod)
Analysis of Variance Table

Response: Yield

Terms added sequentially (first to last)
          Df Sum of Sq  Mean Sq F Value     Pr(F)
Temp       1    1058.0 1058.000   634.8 0.0001371
Conc       1      50.0   50.000    30.0 0.0119669
Caty       1       4.5    4.500     2.7 0.1988922
Temp:Caty  1     200.0  200.000   120.0 0.0016286
Residuals  3       5.0    1.667

402 > plot(mymod)
```



Note that in this data, there is only one observation per cell. This means that the highest order interaction (`Temp:Conc:Caty`) cannot be estimated along with error.

```
402 > mymod _ aov(Yield ~ Temp*Conc*Caty , data = chem)
402 > summary(mymod)
               Df Sum of Sq Mean Sq
          Temp  1    1058.0  1058.0
          Conc  1      50.0    50.0
          Caty  1       4.5     4.5
     Temp:Conc  1       4.5     4.5
     Temp:Caty  1     200.0   200.0
     Conc:Caty  1       0.0     0.0
Temp:Conc:Caty  1       0.5     0.5
```

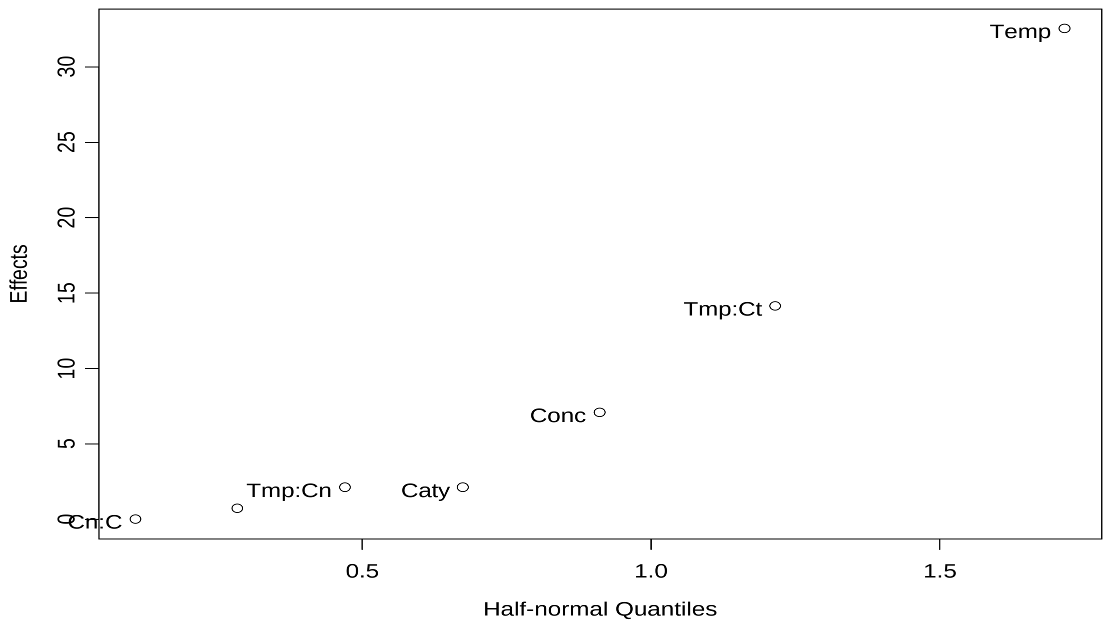
If there were replications in the cells, we *could* estimate `Temp:Conc:Caty` and error at the same time.

All the two-way interactions could be estimated, though as suggested by the interaction plots, they won't all be interesting. As a review (and extension!) of the model notation, here are several ways to fit the model of (main effects and) all two-way interactions.

```
402 > mymod <- aov(Yield ~ Temp + Conc + Caty + Temp:Caty +
+ Temp:Conc + Caty:Conc, data=chem)

402 > mymod <- aov(Yield ~ Temp*Caty + Temp*Conc + Caty*Conc, data=chem)

402 > mymod <- aov(Yield ~ Temp*Conc*Caty - Temp:Conc:Caty,data=chem)

402 > mymod <- aov(Yield ~ (Temp + Conc + Caty)^2, data=chem)

402 > summary(mymod)
          Df Sum of Sq Mean Sq F Value     Pr(F)
     Temp  1    1058.0  1058.0    2116 0.0138374
     Conc  1      50.0    50.0     100 0.0634510
     Caty  1       4.5     4.5       9 0.2048328
Temp:Conc  1       4.5     4.5       9 0.2048328
Temp:Caty  1     200.0   200.0     400 0.0318045
Conc:Caty  1       0.0     0.0       0 1.0000000
Residuals  1       0.5     0.5
```

A "Half-normal plot" is another interesting way to see which interactions and main effects matter and which don't in a model.

```
402 > qqnorm(mymod,label=7)
```



This plots the absolute value of all the estimated parameters (except the intercept) on a qq plot against a half-normal distribution. If all of the effects are insignificant then the estimates are just samples from a half-normal and should line up along a straight line in the half-normal plot. Significant effects show up as high outliers in this plot.

The half-normal plot confirms what the p-values say: `Temp`, `Conc` and `Temp:Caty` are the interesting effects. To retain interpretation of the interaction as "deviation from the additive model", it is **very important** to include `Caty` all lower order terms corresponding to an interaction that we want to keep in a model. Thus, if we want to keep `Temp:Caty` in our model, we need to keep both `Temp` and `Caty` as well, even thought `Caty` is not significant.

3 Fractional Factorial Designs—An Example

Factorial designs are great, but the number of runs required for a full 2^k design increases exponentially in k. When k is large (say bigger than 4 or 5) it is often a good idea to run a fraction of the design—and recover a fraction of the possible information. Hopefully one designs the experiment to recover a useful fraction of the information.

Consider a two-level design with seven variables. A complete factorial design would require $2^7 = 128$ runs, which might be a prohibitive number. These 128 degress of freedom would be partioned into main effects and interactions, as follows

```
average  main  2-fact  3-fact  4-fact  5-fact  6-fact  7-fact
  1        7      21      35      35      21      7        1
```

In many problems one can argue that the high order interactions are probably **not** very interesting. If we were to drop them from the model (assume they are zero!), it would be very much like assuming that the higher order terms of a Taylor Series expansion are zero. (Actually, a Fourier series expansion is the better model, but you get the idea). Whether this is a good approximation to reality depends on the situation, but it often does work well.

The aim of a fractional factorial design is to run the experiment so that the high order interactions (which we might assume to be zero) are confounded with things that we would like to know.

The best way to proceed is via an example.

First lets generate the design and add the data. This lets us illustrate some other SPLUS tricks along the way.

```
402 > reactor <- fac.design(c(2,2,2,2,2))
402 > attach(reactor, pos=1)
402 > Resp <- c(61,53,63,61,53,56,54,61,69,61,94,93,66,60,95,98,
+ 56,63,70,65,59,55,67,65,44,45,78,77,49,42,81,82)
402 > length(Resp)
[1] 32
402 > detach(1, save="newreact")
402 > reactor <- newreact
402 > rm(newreact)
```

The variables are feed rate (10, 15), catalyst (1%, 2%), agitation rate (100,120), temp (140, 180), concentration (3%, 6%) and the response is the percentage of the ingredients that reacted.

Now lets just print out the data to look at.

```
402 > reactor
    A  B  C  D  E Resp
 1 A1 B1 C1 D1 E1   61
 2 A2 B1 C1 D1 E1   53
 3 A1 B2 C1 D1 E1   63
 4 A2 B2 C1 D1 E1   61
 5 A1 B1 C2 D1 E1   53
 6 A2 B1 C2 D1 E1   56
 7 A1 B2 C2 D1 E1   54
 8 A2 B2 C2 D1 E1   61
 9 A1 B1 C1 D2 E1   69
10 A2 B1 C1 D2 E1   61
11 A1 B2 C1 D2 E1   94
12 A2 B2 C1 D2 E1   93
13 A1 B1 C2 D2 E1   66
14 A2 B1 C2 D2 E1   60
15 A1 B2 C2 D2 E1   95
16 A2 B2 C2 D2 E1   98
17 A1 B1 C1 D1 E2   56
18 A2 B1 C1 D1 E2   63
19 A1 B2 C1 D1 E2   70
20 A2 B2 C1 D1 E2   65
21 A1 B1 C2 D1 E2   59
22 A2 B1 C2 D1 E2   55
23 A1 B2 C2 D1 E2   67
24 A2 B2 C2 D1 E2   65
25 A1 B1 C1 D2 E2   44
26 A2 B1 C1 D2 E2   45
27 A1 B2 C1 D2 E2   78
28 A2 B2 C1 D2 E2   77
29 A1 B1 C2 D2 E2   49
30 A2 B1 C2 D2 E2   42
31 A1 B2 C2 D2 E2   81
32 A2 B2 C2 D2 E2   82
```

Now suppose that we had not run the entire experiment, but only half of the runs, in particular, runs

```
402 > runs
 [1]  2  3  5  8  9 12 14 15 17 20 22 23 26 27 29 32
```

So our data might have been

```
402 > half.reactor <- reactor[runs,]
402 > half.reactor
```

```
    A  B  C  D  E Resp
 2 A2 B1 C1 D1 E1   53
 3 A1 B2 C1 D1 E1   63
 5 A1 B1 C2 D1 E1   53
 8 A2 B2 C2 D1 E1   61
 9 A1 B1 C1 D2 E1   69
12 A2 B2 C1 D2 E1   93
14 A2 B1 C2 D2 E1   60
15 A1 B2 C2 D2 E1   95
17 A1 B1 C1 D1 E2   56
20 A2 B2 C1 D1 E2   65
22 A2 B1 C2 D1 E2   55
23 A1 B2 C2 D1 E2   67
26 A2 B1 C1 D2 E2   45
27 A1 B2 C1 D2 E2   78
29 A1 B1 C2 D2 E2   49
32 A2 B2 C2 D2 E2   82
```

This is called a 2^{5-1} design, since it accomodates 2 levels each of 5 variables but only half of the 32 runs of a full 2^5 design were performed.

How would we analyse these data? First off we might like to know what effects are "aliased" with what other effects. *Aliasing* means that the the column of the design matrix (in the linear regression model underlying the ANOVA) corresponding to one effect is linearly dependent on the column of the design matrix for another; two effects that are aliased cannot be separately estimated without an additional linear constraint (like setting one to zero, as we will do). SPLUS can tell us the alias pattern for these data.

```
402 > alias(half.reactor, ~ (A+B+C+D+E)^4)
Model
 ~ (A + B + C + D + E)^4
Complete
        (Intercept)  A  B  C  D  E A:B A:C A:D A:E B:C B:D B:E C:D C:E D:E
  A:B:C                                                                  1
  A:B:D                                                              1
  A:B:E                                                          1
  A:C:D                                                      1
  A:C:E                                                  1
  A:D:E                                              1
  B:C:D                                          1
  B:C:E                                      1
  B:D:E                                  1
  C:D:E                              1
A:B:C:D                          1
A:B:C:E                       1
A:B:D:E                    1
A:C:D:E                 1
B:C:D:E              1
```

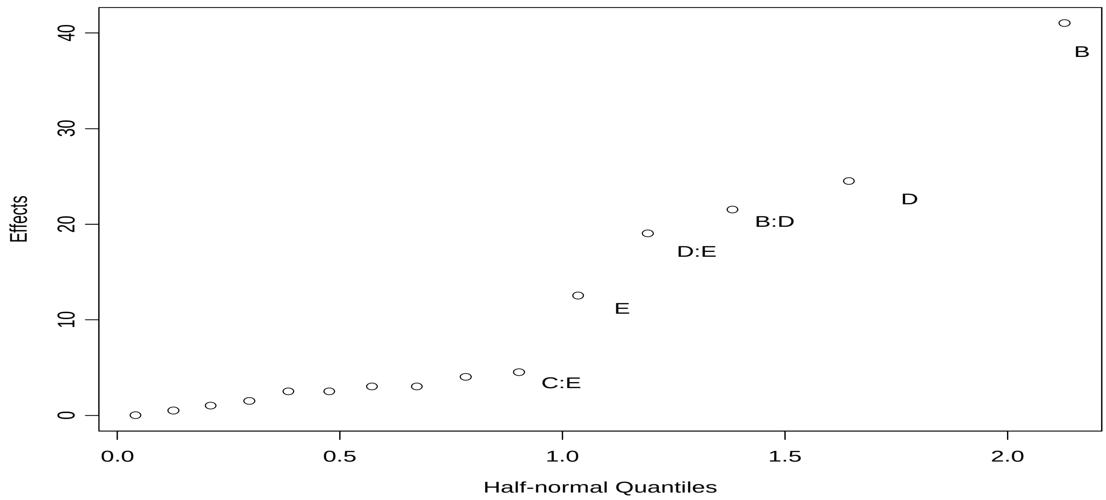
It is important to note that the main effects and two-factor interactions are aliased with three- and four- factor interactions. This is a very desirable type of design (but how did we generate it? **HINT:** Factors A, B, C, and D are given a complete 2^4 factorial design; if you let A1 be a $+$, A2 be a $-$, B1 be a $+$, B2 be a $-$, etc., then the levels of E are the products of the levels of A, B, C and D. In other words "E=ABCD" defines this design).

What about a quick analysis of these data?

Fitting the model of all 2-way interactions,

```
402 > mymod _ aov(Resp ~ (A+B+C+D+E)^2,data=half.reactor)
402 > summary(mymod)
    Df Sum of Sq Mean Sq
  A  1     16.00   16.00
  B  1   1681.00 1681.00
  C  1      0.00    0.00
  D  1    600.25  600.25
  E  1    156.25  156.25
A:B  1      9.00    9.00
A:C  1      1.00    1.00
A:D  1      2.25    2.25
A:E  1      6.25    6.25
B:C  1      9.00    9.00
B:D  1    462.25  462.25
B:E  1      6.25    6.25
C:D  1      0.25    0.25
C:E  1     20.25   20.25
D:E  1    361.00  361.00
402 > qqnorm(mymod,label=T)
```

one quickly sees that not all 2-way interactions can be fitted, but the half normal plot



suggests that B:D and D:E are the important interactions; we will include all the main effects to see what they look like as well.

```
402 > attach(half.reactor, pos=1)
402 > mymod <- aov(Resp ~ (A+B+C+D+E) + B*D + D*E )
402 > anova(mymod)
Analysis of Variance Table

Response: Resp

Terms added sequentially (first to last)
          Df Sum of Sq  Mean Sq  F Value     Pr(F)
A          1     16.00   16.000   2.3594 0.1630778
B          1   1681.00 1681.000 247.8894 0.0000003
C          1      0.00    0.000   0.0000 1.0000000
D          1    600.25  600.250  88.5161 0.0000134
E          1    156.25  156.250  23.0415 0.0013553
B:D        1    462.25  462.250  68.1659 0.0000348
D:E        1    361.00  361.000  53.2350 0.0000842
Residuals  8     54.25    6.781
402 >
```

What would this have looked like with the full data???

```
402 > anova(mymod)
Analysis of Variance Table

Response: Resp

Terms added sequentially (first to last)
          Df Sum of Sq  Mean Sq  F Value     Pr(F)
A          1    15.125   15.125   1.3432 0.2578723
B          1  3042.000 3042.000 270.1499 0.0000000
C          1     3.125    3.125   0.2775 0.6031664
D          1   924.500  924.500  82.1018 0.0000000
E          1   312.500  312.500  27.7521 0.0000211
B:D        1  1404.500 1404.500 124.7290 0.0000000
D:E        1   968.000  968.000  85.9648 0.0000000
Residuals 24   270.250   11.260
```

The answer is only slightly different!