- As you might imagine, as the number of factors becomes large, *interpreting* higher-way interactions -- that is, figuring out what they mean -- becomes more and more difficult. For this reason, sometimes the higher-order interactions are deliberately omitted from the full model in big experimental designs; they are never tested. Is this reasonable? Most of my answers are just elaborate ways to say I don't know.

More than two values for an independent variable

Regardless of how many factors we have, or how many levels there are in each factor, we could always form a combination variable -- that is, a single categorical independent variable whose values represent all the combinations of independent variable values in the factorial design. We have seen that in a two-by-two design, the tests for both main effects and the interaction resolve themselves into tests for single contrasts -- contrasts of the means of the combination variable. When independent variables have more than two values, the same thing is true, except that tests for main effects and interactions appear as test for *collections* of contrasts on the combination variable.

It is useful to pursue this principle in detail, for three reasons.

- First, it thinking of an interaction as a collection of contrasts can really help you understand what an interaction is.
- Second, once you have seen the tests for main effects and interactions as collections of contrasts, you can easily compose a test for any collection of contrasts that is of interest.
- Third, seeing main effects and interactions in terms of contrasts makes it easy to see how they can be modified to become Bonferroni or Scheffe follow-ups to initial significant one-way ANOVA on the combination variable --- if you choose to follow this conservative data analytic strategy.

We'll start with an example.

The seeds of the canola plant yield a high-quality cooking oil. Canola is one of Canada's biggest cash crops. But each year, millions of dollars are lost because of a fungus that kills canola plants. Or is it just one fungus? All this stuff looks the same. It's a nasty black rot that grows fastest under moist, warm conditions. It looks quite a bit like the fungus that grows in between shower tiles.

A team of botanists recognized that although the fungus may look the same, there are actually several different kinds that are genetically distinct. There are also quite a few strains of canola plant, so the questions arose

- Are some strains of fungus more aggressive than others? That is, do they grow faster and overwhelm the plant's defenses faster?
- Are some strains of canola plant more vulnerable to infection than others?
- Are some strains of fungus more dangerous to certain strains of plant and less dangerous to others?

These questions can be answered directly by looking at main effects and the interaction, so a factorial experiment was designed in which canola plants of three different varieties were randomly selected to be infected with one of six genetically different types of fungus. The way they did it was to scrape a little patch at the base of the plant, and wrap the wound with a moist band-aid that had some fungus on it. Then the plant was placed in a very moist dark environment for three days. After three days the bandage was removed and the plant was put in a commercial greenhouse. On each of 14 consecutive days, various measurements were made on the plant. Here, we will be concerned with lesion length, the length of the fungus patch on the plant, measured in millimeters.

The dependent variable will be mean lesion length; the mean is over the 14 daily lesion length measurements for each plant. The independent variables are Cultivar (type of canola plant) and MCG (type of fungus). Type of plant is called cultivar because the fungus grows (is "cultivated") on the plant. MCG stands for "Mycelial Compatibility Group." This strange name comes from the way that the botanists decided whether two types of fungus were genetically distinct. The would grow two samples on the same dish in a nutrient solution, and if the two fungus patches stayed separate, they were genetically different. If they grew together into a single patch of fungus (that is, they were compatible), then they were genetically identical. Apparently, this phenomenon is well established.

Here is the SAS program `appgreen1.sas`. As usual, the entire program is listed first. Then pieces of the program are repeated, together with pieces of output and discussion.

```
/* appgreen1.sas */
%include 'gh91read.sas';
options pagesize=100;
proc freq;
     tables plant*mcg /norow nocol nopercent;
proc glm;
     class plant mcg;
     model meanlng = plant|mcg;
     means plant|mcg;
proc tabulate;
     class mcg plant;
     var meanlng ;
     table (mcg all),(plant all) * (mean*meanlng);

/* Replicate tests for main effects and interactions, using contrasts on a
   combination variable. This is the hard way to do it, but if you can do
   this, you understand interactions and you can test any collection of
   contrasts. The definition of the variable combo could have been in
   gh91read.sas   */

data slime;
     set mould; /* mould was created by ghread91.sas */
     if         plant=1 and mcg=1 then combo =  1;
        else if plant=1 and mcg=2 then combo =  2;
        else if plant=1 and mcg=3 then combo =  3;
        else if plant=1 and mcg=7 then combo =  4;
        else if plant=1 and mcg=8 then combo =  5;
        else if plant=1 and mcg=9 then combo =  6;
        else if plant=2 and mcg=1 then combo =  7;
        else if plant=2 and mcg=2 then combo =  8;
        else if plant=2 and mcg=3 then combo =  9;
        else if plant=2 and mcg=7 then combo = 10;
        else if plant=2 and mcg=8 then combo = 11;
        else if plant=2 and mcg=9 then combo = 12;
        else if plant=3 and mcg=1 then combo = 13;
        else if plant=3 and mcg=2 then combo = 14;
        else if plant=3 and mcg=3 then combo = 15;
        else if plant=3 and mcg=7 then combo = 16;
        else if plant=3 and mcg=8 then combo = 17;
        else if plant=3 and mcg=9 then combo = 18;
     label combo = 'Plant-MCG Combo';
```

```
/* Getting main effects and the interaction with CONTRAST statements */
proc glm;
     class combo;
     model meanlng = combo;
     contrast 'Plant Main Effect'
         combo 1  1  1  1  1  1  -1 -1 -1 -1 -1 -1   0  0  0  0  0  0,
         combo 0  0  0  0  0  0   1  1  1  1  1  1  -1 -1 -1 -1 -1 -1;
     contrast 'MCG Main Effect'
         combo 1 -1  0  0  0  0   1 -1  0  0  0  0   1 -1  0  0  0  0,
         combo 0  1 -1  0  0  0   0  1 -1  0  0  0   0  1 -1  0  0  0,
         combo 0  0  1 -1  0  0   0  0  1 -1  0  0   0  0  1 -1  0  0,
         combo 0  0  0  1 -1  0   0  0  0  1 -1  0   0  0  0  1 -1  0,
         combo 0  0  0  0  1 -1   0  0  0  0  1 -1   0  0  0  0  1 -1;
     contrast 'Plant by MCG Interaction'
         combo -1  1  0  0  0  0   1 -1  0  0  0  0   0  0  0  0  0  0,
         combo  0  0  0  0  0  0  -1  1  0  0  0  0   1 -1  0  0  0  0,
         combo  0 -1  1  0  0  0   0  1 -1  0  0  0   0  0  0  0  0  0,
         combo  0  0  0  0  0  0   0 -1  1  0  0  0   0  1 -1  0  0  0,
         combo  0  0 -1  1  0  0   0  0  1 -1  0  0   0  0  0  0  0  0,
         combo  0  0  0  0  0  0   0  0 -1  1  0  0   0  0  1 -1  0  0,
         combo  0  0  0 -1  1  0   0  0  0  1 -1  0   0  0  0  0  0  0,
         combo  0  0  0  0  0  0   0  0  0 -1  1  0   0  0  0  1 -1  0,
         combo  0  0  0  0 -1  1   0  0  0  0  1 -1   0  0  0  0  0  0,
         combo  0  0  0  0  0  0   0  0  0  0 -1  1   0  0  0  0  1 -1;

/* proc reg's test statement may be easier, but first we need to
   make 16 dummy variables for cell means coding. This will illustrate
   arrays and loops, too */

data yucky;
     set slime;
     array mu{18} mu1-mu18;
     do i=1 to 18;
          if combo=. then mu{i}=.;
          else if combo=i then mu{i}=1;
          else mu{i}=0;
     end;

proc reg;
     model meanlng = mu1-mu18 / noint;
     alleq:   test mu1=mu2=mu3=mu4=mu5=mu6=mu7=mu8=mu9=mu10=mu11=mu12
                   = mu13=mu14=mu15=mu16=mu17=mu18;

     plant:   test mu1+mu2+mu3+mu4+mu5+mu6    = mu7+mu8+mu9+mu10+mu11+mu12,
                   mu7+mu8+mu9+mu10+mu11+mu12 = mu13+mu14+mu15+mu16+mu17+mu18;

     fungus:  test mu1+mu7+mu13 = mu2+mu8+mu14 = mu3+mu9+mu15
                   = mu4+mu10+mu16 = mu5+mu11+mu17 = mu6+mu12+mu18;

     p_by_f:  test mu2-mu1=mu8-mu7=mu14-mu13,
                   mu3-mu2=mu9-mu8=mu15-mu14,
                   mu4-mu3=mu10-mu9=mu16-mu15,
                   mu5-mu4=mu11-mu10=mu17-mu16,
                   mu6-mu5=mu12-mu11=mu18-mu17;
```

```
/* Now illustrate effect coding, with the interaction represented by a
   collection of product terms.  */

data nasty;
     set yucky;
     /* Two dummy variables for plant */
        if plant=. then p1=.;
        else if plant=1 then p1=1;
        else if plant=3 then p1=-1;
        else p1=0;
     if plant=. then p2=.;
        else if plant=2 then p2=1;
        else if plant=3 then p2=-1;
        else p2=0;
     /* Five dummy variables for mcg */
     if mcg=. then f1=.;
        else if mcg=1 then f1=1;
        else if mcg=9 then f1=-1;
        else f1=0;
     if mcg=. then f2=.;
        else if mcg=2 then f2=1;
        else if mcg=9 then f2=-1;
        else f2=0;
     if mcg=. then f3=.;
        else if mcg=3 then f3=1;
        else if mcg=9 then f3=-1;
        else f3=0;
     if mcg=. then f4=.;
        else if mcg=7 then f4=1;
        else if mcg=9 then f4=-1;
        else f4=0;
     if mcg=. then f5=.;
        else if mcg=8 then f5=1;
        else if mcg=9 then f5=-1;
        else f5=0;
     /* Product terms for interactions */
        p1f1 = p1*f1; p1f2=p1*f2 ; p1f3=p1*f3 ; p1f4=p1*f4; p1f5=p1*f5;
        p2f1 = p2*f1; p2f2=p2*f2 ; p2f3=p2*f3 ; p2f4=p2*f4; p2f5=p2*f5;

proc reg;
     model meanlng = p1 -- p2f5;
     plant:  test p1=p2=0;
     mcg:    test f1=f2=f3=f4=f5=0;
     p_by_f: test p1f1=p1f2=p1f3=p1f4=p1f5=p2f1=p2f2=p2f3=p2f4=p2f5 = 0;
```

The SAS program starts with a `%include` statement that reads `ghread91.sas`. The file `ghread91.sas` consists of a single big data step. We'll skip it, because all we really need are the two independent variables `plant` and `mcg`, and the dependent variable `meanlng`.

Just to see what we've got, we do a proc freq to show the sample sizes.

```
proc freq;
     tables plant*mcg /norow nocol nopercent;
```

and we get

TABLE OF PLANT BY MCG

PLANT(Type of Plant) MCG(Mycelial Compatibility Group)

Frequency	1	2	3	7	8	9	Total
GP159	6	6	6	6	6	6	36
HANNA	6	6	6	6	6	6	36
WESTAR	6	6	6	6	6	6	36
Total	18	18	18	18	18	18	108

So it's a nice 3 by 6 factorial design, with 6 plants in each treatment combination. The proc glm for analyzing this is straightforward. Again, we get all main effects and interactions for the factor names separated by vertical bars.

```
proc glm;
     class plant mcg;
     model meanlng = plant|mcg;
     means plant|mcg;
```

And the output is

General Linear Models Procedure
Class Level Information

Class	Levels	Values
PLANT	3	GP159 HANNA WESTAR
MCG	6	1 2 3 7 8 9

Number of observations in data set = 108

General Linear Models Procedure

Dependent Variable: MEANLNG Average Lesion length

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	17	328016.87350	19295.11021	19.83	0.0001
Error	90	87585.62589	973.17362		
Corrected Total	107	415602.49939			

R-Square	C.V.	Root MSE	MEANLNG Mean
0.789256	48.31044	31.195731	64.573479

Source	DF	Type I SS	Mean Square	F Value	Pr > F
PLANT	2	221695.12747	110847.56373	113.90	0.0001
MCG	5	58740.26456	11748.05291	12.07	0.0001
PLANT*MCG	10	47581.48147	4758.14815	4.89	0.0001

Source	DF	Type III SS	Mean Square	F Value	Pr > F
PLANT	2	221695.12747	110847.56373	113.90	0.0001
MCG	5	58740.26456	11748.05291	12.07	0.0001
PLANT*MCG	10	47581.48147	4758.14815	4.89	0.0001

Notice that the Type I and Type III tests are the same. This always happens when the sample sizes are equal.

General Linear Models Procedure

Level of PLANT	N	MEANLNG Mean	MEANLNG SD
GP159	36	14.055159	12.1640757
HANNA	36	55.700198	30.0137912
WESTAR	36	123.965079	67.0180440

Level of MCG	N	MEANLNG Mean	MEANLNG SD
1	18	41.4500000	33.6183462
2	18	92.1333333	78.3509451
3	18	87.5857143	61.7086751
7	18	81.7603175	82.6711755
8	18	50.8579365	39.3417859
9	18	33.6535714	39.1480830

Level of PLANT	Level of MCG	N	MEANLNG Mean	MEANLNG SD
GP159	1	6	12.863095	12.8830306
GP159	2	6	21.623810	17.3001296
GP159	3	6	14.460714	7.2165396
GP159	7	6	17.686905	16.4258441
GP159	8	6	8.911905	7.3162618
GP159	9	6	8.784524	6.5970501
HANNA	1	6	45.578571	26.1430472
HANNA	2	6	67.296429	30.2424997
HANNA	3	6	94.192857	20.2877876
HANNA	7	6	53.621429	24.8563497
HANNA	8	6	47.838095	12.6419109
HANNA	9	6	25.673810	17.1723150
WESTAR	1	6	65.908333	35.6968616
WESTAR	2	6	187.479762	45.1992178
WESTAR	3	6	154.103571	26.5469183
WESTAR	7	6	173.972619	79.1793105
WESTAR	8	6	95.823810	22.3712022
WESTAR	9	6	66.502381	52.5253101

The main effects are fairly easy to look at, and we definitely construct a plot from the 18 cell means (or copy them into a nicer-looking table. But the following `proc tabulate` prints a table that is much easier to look at.

```
proc tabulate;
     class mcg plant;
     var meanlng ;
     table (mcg all),(plant all) * (mean*meanlng);
```

The syntax of `proc tabulate` is fairly elaborate, and at times it's worth the effort. Any reader who has seen the type of stub-and-banner tables favoured by professional market researchers will be impressed to hear that `proc tabulate` can come close to that. I figured out how to make the table below by looking in the manual. I then promptly forgot the overall principles, because it's not a tool I use a lot -- and the syntax is rather arcane. However, this example is easy to follow if you want to produce good-looking two-way tables of means. Here's the output.

	Type of Plant			
	GP159	HANNA	WESTAR	ALL
	MEAN	MEAN	MEAN	MEAN
	Average Lesion length	Average Lesion length	Average Lesion length	Average Lesion length
Mycelial Compatibility Group				
1	12.86	45.58	65.91	41.45
2	21.62	67.30	187.48	92.13
3	14.46	94.19	154.10	87.59
7	17.69	53.62	173.97	81.76
8	8.91	47.84	95.82	50.86
9	8.78	25.67	66.50	33.65
ALL	14.06	55.70	123.97	64.57

The proc tabulate output makes it easy to graph the means. But before we do so, let's look at the main effects and interactions as collections of contrasts. This will actually make it easier to figure out what the results mean, once we see what they are.

We have a three by six factorial design that looks like this. Population means are shown in the cells. The single-subscript notation encourages us to think of the combination of MCG and cultivar as a single categorical independent variable with 18 categories.

	MCG (Type of Fungus)					
Cultivar (Type of Plant)	1	2	3	7	8	9
GP159	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6
Hanna	μ_7	μ_8	μ_9	μ_{10}	μ_{11}	μ_{12}
Westar	μ_{13}	μ_{14}	μ_{15}	μ_{16}	μ_{17}	μ_{18}

Next is the part of the SAS program that creates the combination variable. Notice that it involves a data step that comes *after* the `proc glm`. This usually doesn't happen. I did it by creating a new data set called `slime` that starts by being identical to `mould`, which was created in the file `gh91read.sas`. The `set` command is used to read in the data set `mould`, and then we start from there. This is done just for teaching purposes. Ordinarily, I would not create multiple data sets that are mostly copies of each other. I'd put the whole thing in one data step. Here's the code.

```
data slime;
     set mould; /* mould was created by ghread91.sas */
     if         plant=1 and mcg=1 then combo =  1;
        else if plant=1 and mcg=2 then combo =  2;
        else if plant=1 and mcg=3 then combo =  3;
        else if plant=1 and mcg=7 then combo =  4;
        else if plant=1 and mcg=8 then combo =  5;
        else if plant=1 and mcg=9 then combo =  6;
        else if plant=2 and mcg=1 then combo =  7;
        else if plant=2 and mcg=2 then combo =  8;
        else if plant=2 and mcg=3 then combo =  9;
        else if plant=2 and mcg=7 then combo = 10;
        else if plant=2 and mcg=8 then combo = 11;
        else if plant=2 and mcg=9 then combo = 12;
        else if plant=3 and mcg=1 then combo = 13;
        else if plant=3 and mcg=2 then combo = 14;
        else if plant=3 and mcg=3 then combo = 15;
        else if plant=3 and mcg=7 then combo = 16;
        else if plant=3 and mcg=8 then combo = 17;
        else if plant=3 and mcg=9 then combo = 18;
     label combo = 'Plant-MCG Combo';
```

	MCG (Type of Fungus)					
Cultivar (Type of Plant)	1	2	3	7	8	9
GP159	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6
Hanna	μ_7	μ_8	μ_9	μ_{10}	μ_{11}	μ_{12}
Westar	μ_{13}	μ_{14}	μ_{15}	μ_{16}	μ_{17}	μ_{18}

It is clear that the absence of a main effect for Cultivar is the same as

$$\mu_1+\mu_2+\mu_3+\mu_4+\mu_5+\mu_6 = \mu_7+\mu_8+\mu_9+\mu_{10}+\mu_{11}+\mu_{12} = \mu_{13}+\mu_{14}+\mu_{15}+\mu_{16}.$$

There are two equalities here, and they are saying that two contrasts of the eighteen cell means are equal to zero. To see why this is true, consider the first equality

$$\mu_1+\mu_2+\mu_3+\mu_4+\mu_5+\mu_6 = \mu_7+\mu_8+\mu_9+\mu_{10}+\mu_{11}+\mu_{12}$$

Subtracting the quantity on the right-hand side from both sider of the equation, we get

$$\mu_1+\mu_2+\mu_3+\mu_4+\mu_5+\mu_6 - (\mu_7+\mu_8+\mu_9+\mu_{10}+\mu_{11}+\mu_{12}) = 0,$$

and then distributing the minus sign to get rid of the parentheses yields

$$\mu_1+\mu_2+\mu_3+\mu_4+\mu_5+\mu_6-\mu_7-\mu_8-\mu_9-\mu_{10}-\mu_{11}-\mu_{12} = 0. \qquad (4.2)$$

Recall that here, a contrast is a linear combination of the form

$$L = a_1\mu_1 + a_2\mu_2 + ... + a_{18}\mu_{18}.,$$

where the *a* weights add up to zero. Expression (4.2) fits this description, with the first 6 weights equal to one, the next six weights equal to minus one (so they add to zero), and the last 6 weights equal to zero.

The table below gives the weights of the contrasts defining the test for the main effect of plant, one set of weights in each row.

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}
1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1

This is the basis of the first contrast statement in `proc glm`. Notice how the contrasts are separated by commas. Also notice that the variable on which we're doing contrasts (`combo`) has to be repeated.

```
/* Getting main effects and the interaction with CONTRAST statements */
proc glm;
     class combo;
     model meanlng = combo;
     contrast 'Plant Main Effect'
         combo 1  1  1  1  1  1  -1 -1 -1 -1 -1 -1   0  0  0  0  0  0,
         combo 0  0  0  0  0  0   1  1  1  1  1  1  -1 -1 -1 -1 -1 -1;
```

If there is no main effect for MCG, we are saying

$$\mu_1+\mu_7+\mu_{13} = \mu_2+\mu_8+\mu_{14} = \mu_3+\mu_9+\mu_{15} = \mu_4+\mu_{10}+\mu_{16} = \mu_5+\mu_{11}+\mu_{17} = \mu_6+\mu_{12}+\mu_{18}.$$

There are 5 contrasts here, one for each equals sign; there is always an equals sign for each contrast. Here is the table showing the contrasts.

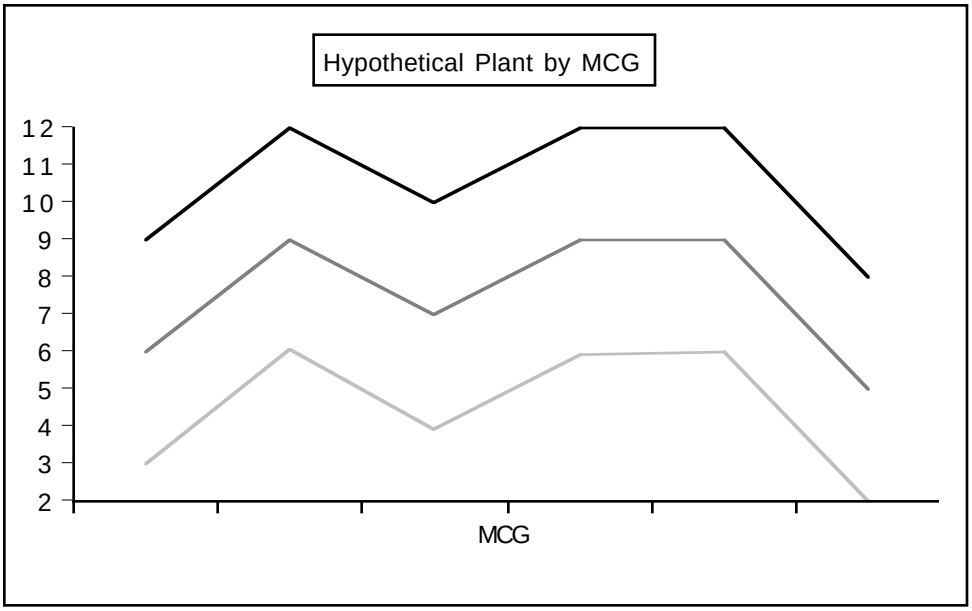
a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}
1	-1	0	0	0	0	1	-1	0	0	0	0	1	-1	0	0	0	0
0	1	-1	0	0	0	0	1	-1	0	0	0	0	1	-1	0	0	0
0	0	1	-1	0	0	0	0	1	-1	0	0	0	0	1	-1	0	0
0	0	0	1	-1	0	0	0	0	1	-1	0	0	0	0	1	-1	0
0	0	0	0	1	-1	0	0	0	0	1	-1	0	0	0	0	1	-1

And here is the corresponding `test` statement in `proc glm`.

```
contrast 'MCG Main Effect'
    combo 1 -1  0  0  0  0   1 -1  0  0  0  0   1 -1  0  0  0  0,
    combo 0  1 -1  0  0  0   0  1 -1  0  0  0   0  1 -1  0  0  0,
    combo 0  0  1 -1  0  0   0  0  1 -1  0  0   0  0  1 -1  0  0,
    combo 0  0  0  1 -1  0   0  0  0  1 -1  0   0  0  0  1 -1  0,
    combo 0  0  0  0  1 -1   0  0  0  0  1 -1   0  0  0  0  1 -1;
```

	MCG (Type of Fungus)					
Cultivar (Type of Plant)	1	2	3	7	8	9
GP159	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6
Hanna	μ_7	μ_8	μ_9	μ_{10}	μ_{11}	μ_{12}
Westar	μ_{13}	μ_{14}	μ_{15}	μ_{16}	μ_{17}	μ_{18}

To compose the Plant by MCG interaction, consider the following hypothetical graph. You can think of the "effect" of MCG as a *profile*, representing a pattern of differences among means. If the three profiles are the same shape for each type of plant -- that is, if they are parallel, the effect of MCG does not depend on the type of plant, and there is no interaction.



For the profiles to be parallel, each set of corresponding line segments must be parallel. To start with the three line segments on the left, the rise represented by $\mu_2-\mu_1$ must equal the rise $\mu_8-\mu_7$, and $\mu_8-\mu_7$ must equal $\mu_{14}-\mu_{13}$. This is two contrasts that equal zero:

$$\mu_2 - \mu_1 - \mu_8 + \mu_7 = 0 \text{ and } \mu_8-\mu_7 -\mu_{14}+\mu_{13} = 0.$$

There are two contrasts for each of the four remaining sets of three line segments, for a total of ten contrasts. They appear directly in the `contrast` statement of `proc glm`. Notice how each row adds to zero; these are *contrasts*, not just linear combinations.

```
contrast 'Plant by MCG Interaction'
    combo -1  1  0  0  0  0   1 -1  0  0  0  0   0  0  0  0  0  0,
    combo  0  0  0  0  0  0  -1  1  0  0  0  0   1 -1  0  0  0  0,
    combo  0 -1  1  0  0  0   0  1 -1  0  0  0   0  0  0  0  0  0,
    combo  0  0  0  0  0  0   0 -1  1  0  0  0   0  1 -1  0  0  0,
    combo  0  0 -1  1  0  0   0  0  1 -1  0  0   0  0  0  0  0  0,
    combo  0  0  0  0  0  0   0  0 -1  1  0  0   0  0  1 -1  0  0,
    combo  0  0  0 -1  1  0   0  0  0  1 -1  0   0  0  0  0  0  0,
    combo  0  0  0  0  0  0   0  0  0 -1  1  0   0  0  0  1 -1  0,
    combo  0  0  0  0 -1  1   0  0  0  0  1 -1   0  0  0  0  0  0,
    combo  0  0  0  0  0  0   0  0  0  0 -1  1   0  0  0  0  1 -1;
```

Now we can compare the tests we get from these contrast statements with what we got from a two-way ANOVA. For easy reference, here is part of the two-way output.

```
Source                     DF      Type III SS     Mean Square  F Value    Pr > F

PLANT                       2     221695.12747    110847.56373   113.90    0.0001
MCG                         5      58740.26456     11748.05291    12.07    0.0001
PLANT*MCG                  10      47581.48147      4758.14815     4.89    0.0001
```

And here is the output from the `contrast` statements.

```
Contrast                   DF      Contrast SS     Mean Square  F Value    Pr > F

Plant Main Effect           2     221695.12747    110847.56373   113.90    0.0001
MCG Main Effect             5      58740.26456     11748.05291    12.07    0.0001
Plant by MCG Interac       10      47581.48147      4758.14815     4.89    0.0001
```

So it worked. Here are some comments.

- Of course this is *not* the way you'd want to test for main effects and interactions. On the contrary, it makes you appreciate all the work that `glm` does for you when you say `model meanlng = plant|mcg;`

- These contrasts are supposed to be an aid to understanding --- understanding what main effects and interactions really are, and understanding how you can test nearly any hypothesis you can think of in a multi-factor design. Almost without exception, what you want to do is test whether some collection of contrasts are equal to zero. Now you can do it, whether the collection you're interested in happens to be standard, or not.

- On the other hand, this was brutal. Even though I am comfortable with high school algebra, the size of the design made specifying those contrasts an unpleasant experience. There is an easier way.

An Easier Way to test Sets of Contrasts in Factorial ANOVA

Because the `test` statement of `proc reg` has a more flexible syntax than the `contrast` statement of `proc glm`, it's a lot easier if you use cell means dummy variable coding, fit a model with no intercept in `proc reg`, and use `test` statements. In the following example, the indicator dummy variables are named `mu1` to `mu18`. This choice makes it possible to directly transcribe statements about the population cell means into test statements. I highly recommend it. Of course if you really hate Greek letters, you could always name them `m1` to `m18` or something.

First, we need to define 18 dummy variables. In general, it's a bit more tedious to define dummy variables than to make a combination variable. Here, I use the combination variable `combo` (which has already been created) to make the task a bit easier -- and also to illustrate the use of arrays and loops in the data step.

```
/* proc reg's test statement may be easier, but first we need to
   make 16 dummy variables for cell means coding. This will illustrate
   arrays and loops, too */

data yucky;
     set slime;
     array mu{18} mu1-mu18;
     do i=1 to 18;
          if combo=. then mu{i}=.;
          else if combo=i then mu{i}=1;
          else mu{i}=0;
     end;

proc reg;
     model meanlng = mu1-mu18 / noint;
     alleq:   test mu1=mu2=mu3=mu4=mu5=mu6=mu7=mu8=mu9=mu10=mu11=mu12
                   = mu13=mu14=mu15=mu16=mu17=mu18;

     plant:   test mu1+mu2+mu3+mu4+mu5+mu6    = mu7+mu8+mu9+mu10+mu11+mu12,
                   mu7+mu8+mu9+mu10+mu11+mu12 = mu13+mu14+mu15+mu16+mu17+mu18;

     fungus:  test mu1+mu7+mu13 = mu2+mu8+mu14 = mu3+mu9+mu15
                   = mu4+mu10+mu16 = mu5+mu11+mu17 = mu6+mu12+mu18;

     p_by_f:  test mu2-mu1=mu8-mu7=mu14-mu13,
                   mu3-mu2=mu9-mu8=mu15-mu14,
                   mu4-mu3=mu10-mu9=mu16-mu15,
                   mu5-mu4=mu11-mu10=mu17-mu16,
                   mu6-mu5=mu12-mu11=mu18-mu17;
```

Looking again at the table of means, it's easy to see how natural the syntax is.

	MCG (Type of Fungus)					
Cultivar (Type of Plant)	1	2	3	7	8	9
GP159	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6
Hanna	μ_7	μ_8	μ_9	μ_{10}	μ_{11}	μ_{12}
Westar	μ_{13}	μ_{14}	μ_{15}	μ_{16}	μ_{17}	μ_{18}

And again, the tests are correct. First, repeat the output from the `contrast` statements of `proc glm` (which matched the `proc glm` two-way ANOVA output).

```
Contrast                 DF      Contrast SS     Mean Square  F Value    Pr > F

Plant Main Effect         2     221695.12747    110847.56373   113.90    0.0001
MCG Main Effect           5      58740.26456     11748.05291    12.07    0.0001
Plant by MCG Interac     10      47581.48147      4758.14815     4.89    0.0001
```

Then, compare output from the test statements of proc reg.

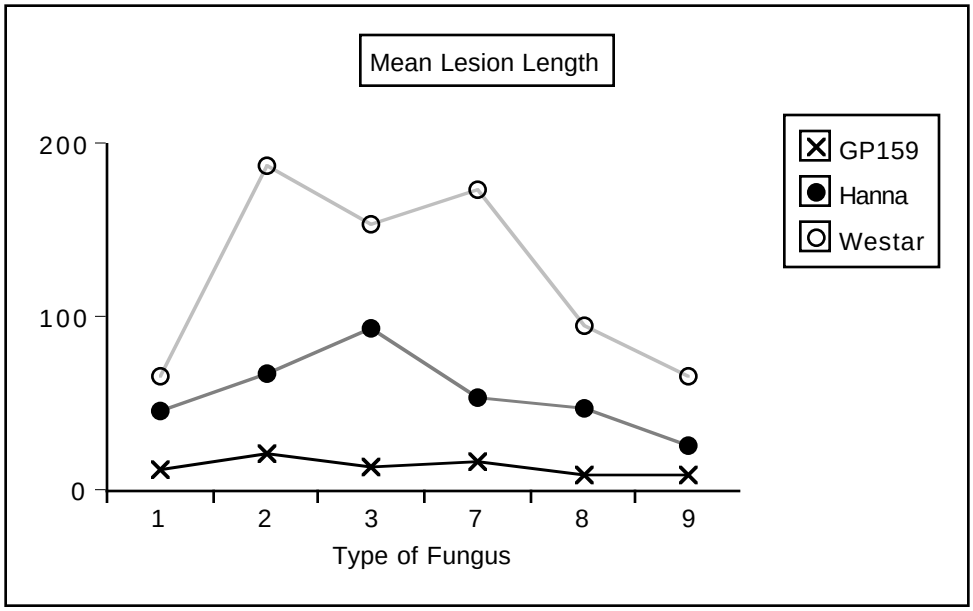
```
Dependent Variable: MEANLNG
Test: ALLEQ     Numerator:  19295.1102  DF:    17   F value:  19.8270
                Denominator:  973.1736  DF:    90   Prob>F:    0.0001


Dependent Variable: MEANLNG
Test: PLANT     Numerator: 110847.5637  DF:     2   F value: 113.9032
                Denominator:  973.1736  DF:    90   Prob>F:    0.0001


Dependent Variable: MEANLNG
Test: FUNGUS    Numerator:  11748.0529  DF:     5   F value:  12.0719
                Denominator:  973.1736  DF:    90   Prob>F:    0.0001


Dependent Variable: MEANLNG
Test: P_BY_F    Numerator:   4758.1481  DF:    10   F value:   4.8893
                Denominator:  973.1736  DF:    90   Prob>F:    0.0001
```

Okay, now we know how to do anything. Finally, it is time to graph the interaction, and find out what these results mean!



First, we see a sizable and clear main effect for Plant. In fact, going back to the analysis of variance summary tables and dividing the Sum of Squares explained by Plant by the Total Sum of Squares, we observe that Plant explains around 53% of the variation in mean lesion length. That's huge. We will definitely want to look at pairwise comparisons of marginal means, too; we'll get back to this later.

Looking at the pattern of means, it's clear that while the main effect of fungus type is statistically significant, this is not something that should be interpreted, because which one is best (worst) depends on the type of plant. That is, we need to look at the interaction.

The profiles really look different. In particular, GP159 not only has a smaller average lesion length, but it seems to exhibit less responsiveness to different strains of fungus. A test for the equality of μ_1 through μ_6 would be valuable. Pairwise comparisons of the 6 means for Hanna and the 6 means for Westar look promising, too.

A Brief Consideration of Multiple Comparisons

The mention of pairwise comparisons brings up the issue of formal multiple comparison follow-up tests for this problem. The way people often do follow-up tests for factorial designs is to make a combination variable and then do all pairwise comparisons. It seems like they do this because they think it's the only thing the software will let them do. Certainly it's better than nothing. Some comments:

With SAS, pairwise comparisons of cell means are *not* the only thing you can do. `Proc glm` will do all pairwise comparisons of *marginal* means quite easily. This means it's easy to follow up a significant and meaningful main effect.

For the present problem, there are 120 possible pairwise comparisons of the 16 cell means. If we do all these as one-at-a-time tests, the chances of false significance are certainly mounting. There is a strong case here for doing multiple comparisons.

Since the sample sizes are equal, Tukey tests are most powerful for all pairwise comparisons. But it's not so simple. Pairwise comparisons within plants (for example, comparing the 6 means for Westar) are interesting, and pairwise comparisons within fungus types (for example, comparison of Hanna, Westar and GP159 for fungus Type 1) are interesting, but the remaining 57 pairwise comparisons are a lot less so.

Also, pairwise comparisons of cell means are not all we want to do. We've already mentioned the need for pairwise comparisons of the marginal means for plants, and we'll soon see that other, less standard comparisons are of interest.

Everything we need to do will involve testing collections of contrasts. The approach we'll take is to do everything as a one-at-a-time custom test initially, and then figure out how we should correct for the fact that we've done a lot of tests.

It's good to be guided by the data. Here we go. The analyses will be done in the SAS program `appgreen2.sas`. As usual, the entire program is given first. But you should be aware that the program was written one piece at a time and executed many times, with later analyses being suggested by the earlier ones.

The program starts by reading in the file `gh91bread.sas`, which is just `gh91read.sas` with the additional variables defined (especially `combo` and `mu1` through `mu18`) that were defined in `appgreen1.sas`.

```
/* appgreen2.sas:  */
%include 'gh91bread.sas';
options pagesize=100;


proc glm;
     title 'Repeating initial Plant by MCG ANOVA, full design';
     class plant mcg;
     model meanlng = plant|mcg;
     means plant|mcg;

/*  A.  Pairwise comparisons of marginal means for plant, full design
    B.  Test all GP159 means equal, full design
    C.  Test profiles for Hanna & Westar parallel, full design         */

proc reg;
     model meanlng = mu1-mu18 / noint;
     A_GvsH:   test  mu1+mu2+mu3+mu4+mu5+mu6 = mu7+mu8+mu9+mu10+mu11+mu12;
     A_GvsW:   test  mu1+mu2+mu3+mu4+mu5+mu6 = mu13+mu14+mu15+mu16+mu17+mu18;
     A_HvsW:   test  mu7+mu8+mu9+mu10+mu11+mu12 = mu13+mu14+mu15+mu16+mu17+mu18;
     B_G159eq: test  mu1=mu2=mu3=mu4=mu5=mu6;
     C_HWpar:  test  mu8-mu7=mu14-mu13,  mu9-mu8=mu15-mu14,
                     mu10-mu9=mu16-mu15, mu11-mu10=mu17-mu16,
                     mu12-mu11=mu18-mu17;


/*  D.  Oneway on mcg, GP158 subset  */

data just159;  /* This data set will have just GP159 */
     set mould;
     if plant=1;

proc glm data=just159;
     title 'D.  Oneway on mcg, GP158 subset';
     class mcg;
     model meanlng = mcg;

/*  E.  Plant by MCG, Hanna-Westar subset   */

data hanstar;  /* This data set will have just Hanna and Westar */
     set mould;
     if plant ne 1;

proc glm data=hanstar;
     title 'E.  Plant by MCG, Hanna-Westar subset';
     class plant mcg;
     model meanlng = plant|mcg;
```

```
/*  F.  Plant by MCG followup, Hanna-Westar subset
                  Interaction:  Follow with all pairwise differences of
                     Westar minus Hanna differences  */

proc reg;
     model meanlng = mu7-mu18 / noint;
     F_inter: test   mu13-mu7=mu14-mu8=mu15-mu9
                   = mu16-mu10=mu17-mu11=mu18-mu12;
     F_1vs2:  test   mu13-mu7=mu14-mu8;
     F_1vs3:  test   mu13-mu7=mu15-mu9;
     F_1vs7:  test   mu13-mu7=mu16-mu10;
     F_1vs8:  test   mu13-mu7=mu17-mu11;
     F_1vs9:  test   mu13-mu7=mu18-mu12;
     F_2vs3:  test   mu14-mu8=mu15-mu9;
     F_2vs7:  test   mu14-mu8=mu16-mu10;
     F_2vs8:  test   mu14-mu8=mu17-mu11;
     F_2vs9:  test   mu14-mu8=mu18-mu12;
     F_3vs7:  test   mu15-mu9=mu16-mu10;
     F_3vs8:  test   mu15-mu9=mu17-mu11;
     F_3vs9:  test   mu15-mu9=mu18-mu12;
     F_7vs8:  test   mu16-mu10=mu17-mu11;
     F_7vs9:  test   mu16-mu10=mu18-mu12;
     F_8vs9:  test   mu17-mu11=mu18-mu12;


proc iml; /* Critical values for Scheffe tests */
     interac = finv(.95,5,60) ; print interac;
     oneway = finv(.95,11,60); print oneway;
```

After reading and defining the data with a `%include` statement, the program repeats the initial three by six ANOVA from `appgreen1.sas`. This is just for completeness.

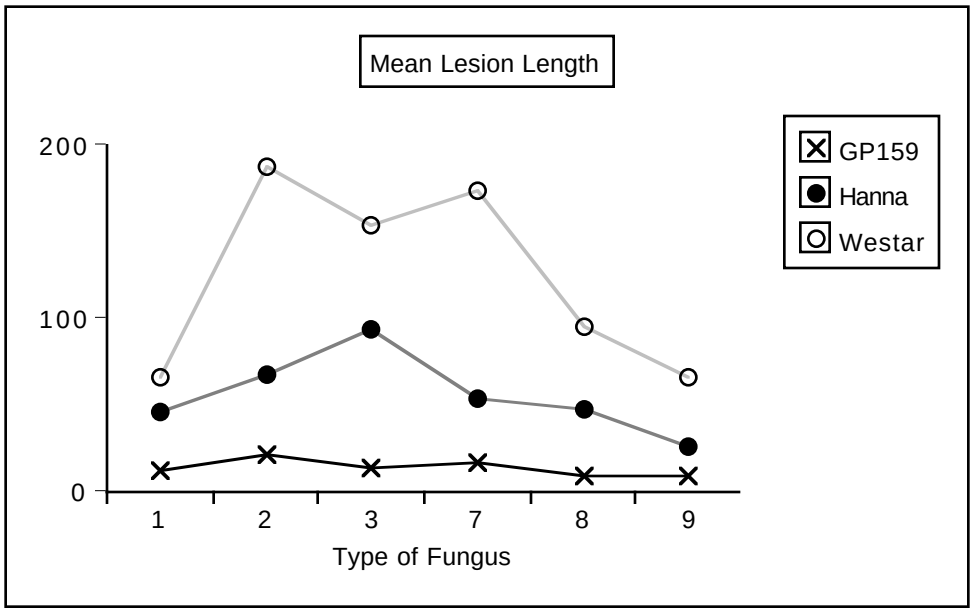
A. It then uses `proc reg` to fit a cell means model, and then tests for all three pairwise differences among Plant means. They are all significantly different from each other, confirming what appears visually in the interaction plot.

```
proc reg;
     model meanlng = mu1-mu18 / noint;
     A_GvsH: test  mu1+mu2+mu3+mu4+mu5+mu6 = mu7+mu8+mu9+mu10+mu11+mu12;
     A_GvsW: test  mu1+mu2+mu3+mu4+mu5+mu6 = mu13+mu14+mu15+mu16+mu17+mu18;
     A_HvsW: test  mu7+mu8+mu9+mu10+mu11+mu12 = mu13+mu14+mu15+mu16+mu17+mu18;
```

```
Dependent Variable: MEANLNG
Test: A_GVSH   Numerator:  31217.5679  DF:    1   F value:  32.0781
               Denominator:  973.1736  DF:   90   Prob>F:    0.0001

Dependent Variable: MEANLNG
Test: A_GVSW   Numerator: 217443.4318  DF:    1   F value: 223.4374
               Denominator:  973.1736  DF:   90   Prob>F:    0.0001

Dependent Variable: MEANLNG
Test: A_HVSW   Numerator:  83881.6915  DF:    1   F value:  86.1940
               Denominator:  973.1736  DF:   90   Prob>F:    0.0001
```



As mentioned earlier, GP159 not only has a smaller average lesion length, but it seems to exhibit less variation in its vulnerability to different strains of fungus. Part of the significant interaction must come from this, and part from differences in the profiles of Hanna and Westar. Two questions arise:

1. Are μ_1 through μ_6 (the means for GP159) actually different from each other?
2. Are the profiles for Hanna and Westar different?

There are two natural ways to address these questions. The naive way is to subset the data --- that is, do a one-way ANOVA to compare the 6 means for GP159, and a two-way (2 by 6) on the Hanna-Westar subset. In the latter analysis, the interaction of Plant by MCG would indicate whether the two profiles were different.

A more sophisticated approach is not to subset the data, but to recognize that both questions can be answered by testing collections of contrasts of the entire set of 18 means; it's easy to do with the `test` statement of `proc reg`.

The advantage of the sophisticated approach is this. Remember that the model specifies a conditional normal distribution of the dependent variable for each combination of independent variable values (in this case there are 18 combinations of independent variable values), and that each conditional distribution has the *same variance.* The test for, say, the equality of μ_1 through μ_6 would use only $\bar{Y}_1$ through $\bar{Y}_6$ (that is, just GP159 data) to estimate the 5 contrasts involved, but it would use *all* the data to estimate the common error variance. From both a commonsense viewpoint and the deepest possible theoretical viewpoint, it's better not to throw information away. This is why the sophisticated approach should be better.

However, this argument is convincing only if it's really true that the dependent variable has the same variance for every combination of independent variable values. Repeating some output from the `means` command of the very first `proc glm`,

Level of PLANT	Level of MCG	N	MEANLNG Mean	MEANLNG SD
GP159	1	6	12.863095	12.8830306
GP159	2	6	21.623810	17.3001296
GP159	3	6	14.460714	7.2165396
GP159	7	6	17.686905	16.4258441
GP159	8	6	8.911905	7.3162618
GP159	9	6	8.784524	6.5970501
HANNA	1	6	45.578571	26.1430472
HANNA	2	6	67.296429	30.2424997
HANNA	3	6	94.192857	20.2877876
HANNA	7	6	53.621429	24.8563497
HANNA	8	6	47.838095	12.6419109
HANNA	9	6	25.673810	17.1723150
WESTAR	1	6	65.908333	35.6968616
WESTAR	2	6	187.479762	45.1992178
WESTAR	3	6	154.103571	26.5469183
WESTAR	7	6	173.972619	79.1793105
WESTAR	8	6	95.823810	22.3712022
WESTAR	9	6	66.502381	52.5253101

we see that the sample standard deviations for GP159 look quite a bit smaller on average. Without bothering to do a formal test, we have some reason to doubt the equal variances assumption.

It's easy to see *why* GP159 would have less plant-to-plant variation in lesion length. It's so resistant to the fungus that there's just not that much fungal growth, period. So there's less *opportunity* for variation.

Note that the equal variances assumption is essentially just a mathematical convenience. Here, it's clearly unrealistic. But what's the consequence of violating it? It's well known that the equal variance assumption can be safely violated if the cell sample sizes are equal and large. Well, here they're equal, but n=6 is not large. So this is not reassuring.

In general, it's not easy to say HOW the tests will be affected when the equal variance assumption is violated, but for the two particular cases we're interested in here (are the GP159 means equal and are the Hanna and Westar profiles parallel), we can figure it out. Recall Formula (3.3) for the F-test.

$$F = \frac{(SSR_F - SSR_R) / s}{MSE_F} .$$

The denominator --- Mean Squared Error from the full model --- is the estimated population error variance. That's the variance that's supposed to be the same for each conditional distribution. Since

$$MSE_F = \frac{\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2}{n - p} ,$$

and the predicted value $\hat{Y}_i$ is always the cell mean, we can draw the following conclusions.

1. When we test for equality of the GP159 means, using the Hanna-Westar data to help compute MSE will make the denominator of F bigger than it should be -- so F is made smaller, and the test is too conservative.

2. When we test whether the Hanna and Westar profiles are parallel, use of the GP159 data to help compute MSE will make the denominator of F *smaller* than it should be -- so F is made bigger, and the test is not conservative enough. That is, the chance of significance if the effect is absent will be greater than 0.05.

This makes me inclined to favour the "naive" subsetting approach. Because the GP159 means LOOK so equal, and I want them to be equal, I'd like to give the test for difference among them the best possible chance. And because it looks like the profiles for Hanna and Westar are not parallel (and I want them to be non-parallel, because it's more interesting for the effect of Fungus type to depend on type of Plant), I want a more conservative test.

Another argument in favour of subsetting is based on botany rather than statistics. Hanna and Westar are commercial canola crop varieties, but while GP159 is definitely in the canola family, it is more like a hardy weed than a food plant. It's just a different kind of entity, and so analyzing its data separately makes a lot of sense.

You may wonder, if it's so different, why was it included in the design in the first place? Well, taxonomically it's quite similar to Hanna and Westar; really no one knew it would be such a vigorous monster in terms of resisting fungus. That's why people do research -- to find out things they didn't already know.

Anyway, we'll do the analysis both ways -- both the seemingly naive way which is probably better once you think about it, and the sophisticated way that uses the complete set of data for all analyses.

Parts B and C represent the "sophisticated" approach that does not subset the data.

B. Test all GP159 means equal, full design

C. Test profiles for Hanna & Westar parallel, full design

```
proc reg;
     model meanlng = mu1-mu18 / noint;
     A_GvsH:   test  mu1+mu2+mu3+mu4+mu5+mu6 = mu7+mu8+mu9+mu10+mu11+mu12;
     A_GvsW:   test  mu1+mu2+mu3+mu4+mu5+mu6 = mu13+mu14+mu15+mu16+mu17+mu18;
     A_HvsW:   test  mu7+mu8+mu9+mu10+mu11+mu12 = mu13+mu14+mu15+mu16+mu17+mu18;
     B_G159eq: test  mu1=mu2=mu3=mu4=mu5=mu6;
     C_HWpar:  test  mu8-mu7=mu14-mu13,  mu9-mu8=mu15-mu14,
                     mu10-mu9=mu16-mu15, mu11-mu10=mu17-mu16,
                     mu12-mu11=mu18-mu17;


Dependent Variable: MEANLNG
Test: B_G159EQ Numerator:    151.5506  DF:    5   F value:   0.1557
               Denominator:  973.1736  DF:   90   Prob>F:    0.9778

Dependent Variable: MEANLNG
Test: C_HWPAR  Numerator:   5364.0437  DF:    5   F value:   5.5119
               Denominator:  973.1736  DF:   90   Prob>F:    0.0002
```

This confirms the visual impression of no differences among means for GP159, and non-parallel profiles for Hanna and Westar. Now compare the subsetting approach. Notice the creation of SAS data sets with subsets of the data.

D. Oneway on mcg, GP158 subset

E. Plant by MCG, Hanna-Westar subset

```
data just159; /* This data set will have just GP159 */
     set mould;
     if plant=1;

proc glm data=just159;
     title 'D.  Oneway on mcg, GP158 subset';
     class mcg;
     model meanlng = mcg;
```

```
                        D.  Oneway on mcg, GP158 subset                        2
                                                  10:52 Friday, February 22, 2002

                        General Linear Models Procedure

Dependent Variable: MEANLNG   Average Lesion length
                                        Sum of            Mean
Source                    DF          Squares          Square  F Value    Pr > F

Model                      5     757.75319161    151.55063832     1.03    0.4189

Error                     30    4421.01258503    147.36708617

Corrected Total           35    5178.76577664

                   R-Square             C.V.        Root MSE       MEANLNG Mean

                   0.146319         86.37031       12.139485          14.055159


Source                    DF        Type I SS     Mean Square  F Value    Pr > F

MCG                        5     757.75319161    151.55063832     1.03    0.4189

Source                    DF      Type III SS     Mean Square  F Value    Pr > F

MCG                        5     757.75319161    151.55063832     1.03    0.4189
```

This analysis is consistent with what we got without subsetting the data. That is, it does not provide evidence that the means for GP159 are different. But when we didn't subset the data, we had p = 0.9778. This happened exactly because including Hanna and Westar data made MSE larger, F smaller, and hence p bigger.

```
data hanstar; /* This data set will have just Hanna and Westar */
     set mould;
     if plant ne 1;

proc glm data=hanstar;
     title 'E.  Plant by MCG, Hanna-Westar subset';
     class plant mcg;
     model meanlng = plant|mcg;
```

E. Plant by MCG, Hanna-Westar subset 3
10:52 Friday, February 22, 2002

General Linear Models Procedure
Class Level Information

Class	Levels	Values
PLANT	2	HANNA WESTAR
MCG	6	1 2 3 7 8 9

Number of observations in data set = 72

E. Plant by MCG, Hanna-Westar subset 4
10:52 Friday, February 22, 2002

General Linear Models Procedure

Dependent Variable: MEANLNG Average Lesion length

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	11	189445.68433	17222.33494	12.43	0.0001
Error	60	83164.61331	1386.07689		
Corrected Total	71	272610.29764			

R-Square	C.V.	Root MSE	MEANLNG Mean
0.694932	41.44379	37.230054	89.832639

Source	DF	Type I SS	Mean Square	F Value	Pr > F
PLANT	1	83881.691486	83881.691486	60.52	0.0001
MCG	5	78743.774570	15748.754914	11.36	0.0001
PLANT*MCG	5	26820.218272	5364.043654	3.87	0.0042

Source	DF	Type III SS	Mean Square	F Value	Pr > F
PLANT	1	83881.691486	83881.691486	60.52	0.0001
MCG	5	78743.774570	15748.754914	11.36	0.0001
PLANT*MCG	5	26820.218272	5364.043654	3.87	0.0042

=========

The significant interaction indicates that the profiles for Hanna and Westar are non-parallel, confirming the visual impression we got from the interaction plot. But the p-value is larger this time. When all the data were used to calculate the error term, we had p = 0.0002. This is definitely due to the low variation in GP159.

Further analyses will be limited to the Hanna-Westar subset.

Now think of the interaction in a different way. Overall, Hanna is more vulnerable than Westar, but the interaction says that the degree of that greater vulnerability depends on the type of fungus. Look at all pairwise comparisons of the DIFFERENCE between Hanna and Westar. First, verify that the interaction can be expressed this way. Of course it can.

F. Plant by MCG followup, Hanna-Westar subset

All pairwise differences of Westar minus Hanna differences

```
proc reg;
     model meanlng = mu7-mu18 / noint;
     F_inter: test   mu13-mu7=mu14-mu8=mu15-mu9
                   = mu16-mu10=mu17-mu11=mu18-mu12;
     F_1vs2:  test   mu13-mu7=mu14-mu8;
     F_1vs3:  test   mu13-mu7=mu15-mu9;
     F_1vs7:  test   mu13-mu7=mu16-mu10;
     F_1vs8:  test   mu13-mu7=mu17-mu11;
     F_1vs9:  test   mu13-mu7=mu18-mu12;
     F_2vs3:  test   mu14-mu8=mu15-mu9;
     F_2vs7:  test   mu14-mu8=mu16-mu10;
     F_2vs8:  test   mu14-mu8=mu17-mu11;
     F_2vs9:  test   mu14-mu8=mu18-mu12;
     F_3vs7:  test   mu15-mu9=mu16-mu10;
     F_3vs8:  test   mu15-mu9=mu17-mu11;
     F_3vs9:  test   mu15-mu9=mu18-mu12;
     F_7vs8:  test   mu16-mu10=mu17-mu11;
     F_7vs9:  test   mu16-mu10=mu18-mu12;
     F_8vs9:  test   mu17-mu11=mu18-mu12;


Dependent Variable: MEANLNG
Test: F_INTER  Numerator:   5364.0437  DF:    5   F value:   3.8699
               Denominator:  1386.077  DF:   60   Prob>F:    0.0042


Dependent Variable: MEANLNG
Test: F_1VS2   Numerator:  14956.1036  DF:    1   F value:  10.7902
               Denominator:  1386.077  DF:   60   Prob>F:    0.0017


Dependent Variable: MEANLNG
Test: F_1VS3   Numerator:   2349.9777  DF:    1   F value:   1.6954
               Denominator:  1386.077  DF:   60   Prob>F:    0.1979


Dependent Variable: MEANLNG
Test: F_1VS7   Numerator:  15006.4293  DF:    1   F value:  10.8265
               Denominator:  1386.077  DF:   60   Prob>F:    0.0017
```

```
Dependent Variable: MEANLNG
Test: F_1VS8   Numerator:   1147.2776  DF:    1   F value:   0.8277
               Denominator:  1386.077  DF:   60   Prob>F:    0.3666


Dependent Variable: MEANLNG
Test: F_1VS9   Numerator:    630.3018  DF:    1   F value:   0.4547
               Denominator:  1386.077  DF:   60   Prob>F:    0.5027


Dependent Variable: MEANLNG
Test: F_2VS3   Numerator:   5449.1829  DF:    1   F value:   3.9314
               Denominator:  1386.077  DF:   60   Prob>F:    0.0520


Dependent Variable: MEANLNG
Test: F_2VS7   Numerator:      0.0423  DF:    1   F value:   0.0000
               Denominator:  1386.077  DF:   60   Prob>F:    0.9956


Dependent Variable: MEANLNG
Test: F_2VS8   Numerator:   7818.7443  DF:    1   F value:   5.6409
               Denominator:  1386.077  DF:   60   Prob>F:    0.0208


Dependent Variable: MEANLNG
Test: F_2VS9   Numerator:   9445.7674  DF:    1   F value:   6.8147
               Denominator:  1386.077  DF:   60   Prob>F:    0.0114


Dependent Variable: MEANLNG
Test: F_3VS7   Numerator:   5479.5767  DF:    1   F value:   3.9533
               Denominator:  1386.077  DF:   60   Prob>F:    0.0513


Dependent Variable: MEANLNG
Test: F_3VS8   Numerator:    213.3084  DF:    1   F value:   0.1539
               Denominator:  1386.077  DF:   60   Prob>F:    0.6962


Dependent Variable: MEANLNG
Test: F_3VS9   Numerator:    546.1923  DF:    1   F value:   0.3941
               Denominator:  1386.077  DF:   60   Prob>F:    0.5326


Dependent Variable: MEANLNG
Test: F_7VS8   Numerator:   7855.1432  DF:    1   F value:   5.6672
               Denominator:  1386.077  DF:   60   Prob>F:    0.0205
```

```
Dependent Variable: MEANLNG
Test: F_7VS9   Numerator:   9485.7704  DF:    1   F value:   6.8436
               Denominator:  1386.077  DF:   60   Prob>F:    0.0112


Dependent Variable: MEANLNG
Test: F_8VS9   Numerator:     76.8370  DF:    1   F value:   0.0554
               Denominator:  1386.077  DF:   60   Prob>F:    0.8147
```

These analyses are summarized in the table below. Westar-Hanna differences that with the same letter are not significantly different.

MCG	120.Westar-Hanna Difference		
7	120.35	A	
2	120.18	A	
3	59.91	A	B
8	47.98		B
9	40.83		B
1	20.33		B

```
proc iml; /* Critical values for Scheffe tests */
     interac = finv(.95,5,60) ; print interac;
     oneway = finv(.95,11,60); print oneway;
```

```
                INTERAC
              2.3682702


                ONEWAY
              1.9522119
```