Chapter Four: Multiple Regression II

Factorial ANOVA and Related Topics

4.1 A One-way Example

The following is a textbook example taken from Neter et al. (1996). The Kenton Food Company is interested in testing the effect of different package designs on sales. Five grocery stores were randomly assigned to each of four package designs. The package designs used either three or five colours, and either had cartoons or did not. Because of a fire in one of the stores, there were only four stores in the 5-colour cartoon condition.

The dependent variable is sales, defined as number of cases sold. Actually, there are two independent variables: number of colours and presence versus absence of cartoons. But we will initially consider package design as a single categorical independent variable with four values.

Sample Question 4.1 If there is a statistically significant relationship between package design and sales, would we be justified in concluding that differences in package design *caused* differences in sales?

Answer to Sample Question 4.1 Yes, if the study is carried out properly. It's an experimental study.

Sample Question 4.2 Is there a problem with external validity here?

Answer to Sample Question 4.2 It's impossible to tell for sure, but there easily could be. The behaviour of the sales force would have to be controlled somehow. A double blind would be ideal.

The SAS program `appkenton1.sas` does a lot of things, starting with a oneway ANOVA using `proc glm`. The strategy will be to first present the entire program, and then go through it piece by piece and explain what is going on -- with a few major digressions to explain the statistics.

```
/********************** appkenton1.sas **************************/
options linesize=79 pagesize=35 noovp formdlim=' ';
title 'Kenton Oneway Example From Neter et al.';

proc format;
     value pakfmt 1 = '3Colour Cartoon'   2 = '3Col No Cartoon'
                  3 = '5Colour Cartoon'   4 = '5Col No Cartoon';
data food;
     infile 'kenton.dat';
     input package sales;
     label package = 'Package Design'
           sales   = 'Number of Cases Sold';
     format package pakfmt.;

     /* Define ncolours and cartoon */
     if package = 1 or package = 2 then ncolours = 3;
        else if package = 3 or package = 4 then ncolours = 5;
     if package = 1 or package = 3 then cartoon = 'No ';
        else if package = 2 or package = 4 then cartoon = 'Yes';

     /* Indicator Coding for package: Use p4 only if no intercept */
     if package = . then p1 = .; else if package = 1 then p1 = 1;
        else p1 = 0;
     if package = . then p2 = .; else if package = 2 then p2 = 1;
        else p2 = 0;
     if package = . then p3 = .; else if package = 3 then p3 = 1;
        else p3 = 0;
     if package = . then p4 = .; else if package = 4 then p4 = 1;
        else p4 = 0;

/* Basic one-way ANOVA -- well, fairly basic */
proc glm;
     class package;
     model sales = package;
     means package;
     means package / bon tukey scheffe;
     estimate '3Colourvs5Colour' package 1 1 -1 -1 / divisor = 2;

/* Now oneway using proc reg and dummy variables.
   First with intercept */

proc reg;
     model sales = p1 p2 p3;
     ncolour: test p1+p2 = p3; /* 3 vs 5 colours */
```

```
/* Special tests are easier with cell means coding:
   No intercept => No algebra */

proc reg;
     model sales = p1 p2 p3 p4 / noint;
     alleq:   test p1=p2=p3=p4;
     numcol:  test p1+p2 = p3+p4;
     cartoon: test p1+p3 = p2+p4;
     inter1:  test p1-p2 = p3-p4;/* Effect of cartoon depends on ncolours*/
     inter2:  test p1-p3 = p2-p4;/* Effect of ncolours depends on cartoon */
     Y3_N3:   test p1=p2;  /* All pairwise tests */
     Y3_Y5:   test p1=p3;
     Y3_N5:   test p1=p4;
     N3_Y5:   test p2=p3;
     N3_N5:   test p2=p4;
     Y5_N5:   test p3=p4;


/* Actually it's a two-way ANOVA */

proc glm;
     class ncolours cartoon;
     model sales = ncolours|cartoon;

/*   The model statement could have been
     model sales = ncolours cartoon ncolours*cartoon; */
```

`Proc format` provides labels for the package designs. After reading the data in a routine way, `if` statements are used to construct the categorical independent variables `ncolours` and `cartoon`. Notice the extra space in the `'No '` value of the alphanumeric variable `cartoon`. At first I didn't have a space, and `Yes` was truncated to `Ye`.

The indicator dummy variables for `package` will be used to show how the one-way (and two-way) ANOVA is really just a multiple regression. We'll come back to them later, but notice how `proc freq` is used to check that they are defined correctly.

Now we'll look at what the first `proc glm` does.

```
proc glm;
     class package;
     model sales = package;
```

The class statement declares package to be categorical. Without it, proc glm would do a regression with package as a quantitative independent variable. The main F-test for equality of the four means is

```
                        General Linear Models Procedure

Dependent Variable: SALES   Number of Cases Sold
                                    Sum of            Mean
Source                  DF         Squares          Square  F Value    Pr > F

Model                    3    588.22105263    196.07368421    18.59    0.0001
Error                   15    158.20000000     10.54666667
Corrected Total         18    746.42105263

                  R-Square            C.V.        Root MSE         SALES Mean
                  0.788055        17.43042       3.2475632          18.631579
```

We conclude that package design (or, if the study was poorly controlled, some variable confounded with package design) caused a difference in sales. The statement

```
means package;
```

produces mean sales for each value of the variable package:

```
Level of                ------------SALES------------
PACKAGE           N       Mean              SD

3Col No Cartoon   5     13.4000000      3.64691651
3Colour Cartoon   5     14.6000000      2.30217289
5Col No Cartoon   5     27.2000000      3.96232255
5Colour Cartoon   4     19.5000000      2.64575131
```

Such a display is essential for seeing what is going on, but it still does not tell you which means are different from which other means. But before we lose control and start doing all possible t-tests, consider the following.

The Curse of a Thousand t-tests Significance tests are supposed to help screen out random garbage, and help us ignore "trends" that could easily be due to chance. But all the common significance tests are designed in isolation, as if each one were the only test you would ever be doing. The chance of getting significant results when nothing is going on may be about 0.05 (more or less, depending on how well the assumptions are met), but if you do a *lot* of tests on a data set that is purely noise (no true relationships between any independent variable and any dependent variable), the chances of false significance mount up. It's like looking for your birthday in tables of stock market prices. If you look long enough, you will find it.

This problem definitely applies when you have a significant difference among more than two treatment means, and you want to know which ones are different from each other. For example, in an experiment with 10 treatment conditions (this is not an unusually large number, for real experiments), there are 45 pairwise differences among means.

You have to pity the poor scientist who learns about this and is honest enough to take this problem seriously (and let's use the term "scientist" generously to apply to anyone trying to use significance test to learn something about a data set). On one hand, good scientific practice and common sense dictate that if you have gone to the trouble to collect data, you should explore thoroughly and try to learn something from the data. But at the same time, it appears that some stern statistical entity is scolding you, and saying that you're naughty if you peek.

There are two main ways to resolve the dilemma. One is to basically ignore the problem, while perhaps acknowledging that it is there. According to this point of view, well, you're crazy if you don't explore the data. Maybe the true significance level for the entire process is greater than 0.05, but still the use of significance tests is a useful way to decide which results might be real. Nothing's perfect; let's carry on.

The other reaction is to look for ways that significance tests can be modified to allow for the fact that we're doing a lot of them. What we want are methods for holding the chances of false significance to a single low level for a *set* of tests, simultaneously. The general term for such methods is **multiple comparison** procedures. Often, when a significance test (like a one−way ANOVA) tests several things simultaneously and turns out to be significant, multiple comparison procedures are used as a second step, to investigate where the effect came from. In cases like this, the multiple comparisons are called **follow−up tests**, or **post hoc tests**, or sometimes **probing**.

It is generally acknowledged that multiple comparison methods are often helpful (even necessary) for following up significant F−tests in order to see where an effect comes from. There is less agreement on how far the principle should be extended. Personally, I like the idea of limiting the chance of false significance to 0.05 for an entire study — say, for all the tests reported in a scientific paper, and all the ones that were not reported, too. This is a fairly radical view, shared by almost no one. But it can work in practice if you have enough data. More on this later. For now, let's concentrate on following up a significant F test in a one−way analysis of variance.

In the Kenton package design data, there are 4 treatment conditions, and 6 potential pairwise comparisons. The next line in the SAS program

```
means package / bon tukey scheffe;
```

requests three kinds of multiple comparison tests for all pairwise differences among means.

Bonferroni The Bonferroni method is very general, and extends far beyond pairwise comparisons of means. It is a simple correction that can be applied when you are performing multiple tests, and you want to hold the chances of false significance to a single low level for all the tests simultaneously. *It applies when you are testing multiple sets of independent variables, multiple dependent variables, or both.*

The Bonferroni correction consists of simply dividing the desired significance level (that's α, the maximum probability of getting significant results when actually nothing is happening, usually $\alpha = 0.05$) by the number of tests. In a way, you're splitting the alpha equally among the tests you do.

For example, if you want to perform 5 tests at joint significance level 0.05, just do everything as usual, but only declare the results significant at the *joint* 0.05 level if one of the tests gives you $p < 0.01$ $(0.01 = 0.05/5)$. If you want to perform 20 tests at joint significance level 0.05, do the individual tests and calculate individual p−values as usual, but only believe the results of tests that give $p < 0.0025$ $(0.0025 = 0.05/20)$. Say something like "Protecting the 20 tests at joint significance level 0.05 by means of a Bonferroni correction, the difference in reported liking between worms and spinach souffle´ was the only significant food category effect."

The Bonferroni correction is *conservative.* That is, if you perform 20 tests, the probability of getting significance at least once just by chance is *less than* or equal to 0.0025 — usually less. The big advantages of the Bonferroni approach are simplicity and flexibility. It is the only way I know to analyze quantitative and categorical dependent variables simultaneously.

The main disadvantages of the Bonferroni approach are

1. *You have to know how many tests you want to perform in advance, and you have to know what they are.* In a typical data analysis situation, not all the significance tests are planned in advance. The results of one test will give rise to ideas for other tests. If you do this and then apply a Bonferroni correction to all the tests that you happened to do, it no longer protects all the tests simultaneously. On the other hand, you could randomly split your data into an exploratory sample and a replication sample. Test to your heart's content on the first sample. Then, when you think you know what your results are, perform *only* those tests on the replication sample, and protect them simultaneously with a Bonferroni correction. This could be called "Bonferroni–protected cross–validation." It sounds good, eh.

2. *The Bonferroni correction can be too conservative,* especially when the number of tests becomes large. For example, to simultaneously test all 780 correlations in a 40 by 40 correlation matrix at joint $\alpha = 0.05$, you'd only believe correlations with $p < 0.0000641 = 0.05/780$.

Is this "too" conservative? Well, with n = 200 in that 40 by 40 example, you'd need r = 0.27 for significance (compared to r = .14 with no correction). With n = 100 you'd need r = .385, or about 14.8% of one variable explained by another *single* variable. Is this too much to ask? You decide.

Tukey This is Tukey's Honestly Significant Difference (HSD) method. It is not his Least Significant Different (LSD) method, which has a better name but does not really get the job done. Tukey tests apply only to pairwise differences among means in ANOVA. It is based on a deep study of the probability distribution of the difference between the largest sample mean and the smallest sample mean, assuming the population means are in fact all equal.

- If you are interested in all pairwise differences among means and nothing else, and if the sample sizes are equal, Tukey is the best (most powerful) test, period.

- If the sample sizes are unequal, the Tukey tests still get the job of simultaneous protection done, but they are a bit conservative. When sample sizes are unequal, Bonferroni or Scheffé can sometimes be more powerful.

Scheffé It is very easy for me to say too much about Scheffé tests, so this discussion will be limited to testing whether certain linear combinations of treatment means (in a one-way design) are significantly different from zero. Suppose there are *p* treatments (groups, values of the categorical independent variable, whatever you want to call them). A **contrast** is a special kind of linear combination of means in which the weights add up to zero. It has the form

$$L = a_1\bar{Y}_1 + a_2\bar{Y}_2 + \cdots + a_p\bar{Y}_p \text{ ,}$$

where $a_1 + a_2 \cdots + a_p = 0$. The case where all of the *a* values are zero is uninteresting, and is excluded.

By setting $a_1 = 1$ and $a_2 = -1$, we get $L = \bar{Y}_1 - \bar{Y}_2$, so it's easy to see that any pairwise difference is a contrast. Contrasts of sample means estimate the corresponding contrasts of population means, in a perfectly natural way. The Scheffé tests allow testing whether *any* contrast of treatment means differs significantly from zero, with the tests for all possible contrasts simultaneously protected.

When asked for Scheffé, SAS tests all pairwise differences, but *there are infinitely many more contrasts that it does not do, and they are all jointly protected against false significance at the 0.05 level.* You can do as many of them as you want easily, with SAS and a calculator.

It's a miracle. You can do infinitely many tests, all simultaneously protected. You do not have to know what they are in advance. It's an license for unlimited data fishing, at least wthin the class of contrasts of treatment means.

Two more miracles:

- If the initial one-way ANOVA is not significant, it's *impossible* for any of the Scheffé follow-ups to be significant. This is not quite true of Bonferroni or Tukey.
- If the initial one-way ANOVA *is* significant, there *must* be a contrast that is significantly different from zero. It may not be a pairwise difference, you may not think of it, and if you do find one it may not be easy to interpret, but there is at least one out there. Well, actually, there are infinitely many, but they may all be extremely similar to one another.

Here's how you do it. First find the critical value of F for the initial oneway ANOVA (Recall that if a test statistic is greater than the critical value, it's significant). This is part of the default output from `proc glm` when you request Scheffé tests -- or you can use `proc iml`. Then use SAS to compute the usual one-at-a-time F-test for whether the contrast is different from zero. Such "usual" F-tests involve comparing full to reduced models. You can do them with the test statement of `proc reg`, but the `estimate` statement of `proc glm` may be more convenient; use $F = t^2$.

Once you have F, use a calculator to compute

$$F_{sch} = \frac{F}{p-1} . \tag{4.1}$$

If F_{sch} is greater than the critical value, the Scheffé test is significant. Keep doing tests until you run out of ideas.

Notice that dividing by the number of means (minus one) is a kind of penalty for the richness of the infinite family of tests you could do. As soon as Scheffé discovered these tests, people started complaining that the penalty was very severe, and it was too hard to get significance. In my opinion, what's remarkable is not that a license for unlimited fishing is expensive, but that it's for sale at all. You can pay for it by increasing the sample size. Choice of sample size will be discussed later in this chapter.

SAS presents the tests for differences between treatment means in the form of confidence intervals. If the 95% confidence interval does not include zero, the test (Bonferroni, Tukey or Scheffé) is significant at 0.05. Since all three types of follow-up test point to exactly the same conclusions for these data, only the Scheffé will be reproduced here.

```
                    General Linear Models Procedure

                      Scheffe's test for variable: SALES

          NOTE: This test controls the type I experimentwise error rate but
                generally has a higher type II error rate than Tukey's for all
                pairwise comparisons.

             Alpha= 0.05  Confidence= 0.95  df= 15  MSE= 10.54667
                         Critical Value of F= 3.28738

       Comparisons significant at the 0.05 level are indicated by '***'.

                                       Simultaneous            Simultaneous
                                           Lower    Difference     Upper
                 PACKAGE                Confidence    Between   Confidence
                Comparison                 Limit       Means       Limit

 5Col No Cartoon - 5Colour Cartoon         0.859       7.700      14.541   ***
 5Col No Cartoon - 3Colour Cartoon         6.150      12.600      19.050   ***
 5Col No Cartoon - 3Col No Cartoon         7.350      13.800      20.250   ***

 5Colour Cartoon - 5Col No Cartoon       -14.541      -7.700      -0.859   ***
 5Colour Cartoon - 3Colour Cartoon        -1.941       4.900      11.741
 5Colour Cartoon - 3Col No Cartoon        -0.741       6.100      12.941

 3Colour Cartoon - 5Col No Cartoon       -19.050     -12.600      -6.150   ***
 3Colour Cartoon - 5Colour Cartoon       -11.741      -4.900       1.941
 3Colour Cartoon - 3Col No Cartoon        -5.250       1.200       7.650

 3Col No Cartoon - 5Col No Cartoon       -20.250     -13.800      -7.350   ***
```

Notice that the critical value for performing more tests is conveniently provided. Proc `iml`'s

```
proc iml; /* Critical value for Scheffe tests */
     fcrit = finv(.95,3,15); print fcrit;
```

yields the same critical value.

```
          FCRIT
      3.2873821
```

Pairwise differences are not the only contrasts of interest. The first `proc glm` has the line

```
estimate '3Colourvs5Colour' package 1 1 -1 -1 / divisor = 2;
```

Here, we are directly providing the *a* weights of the contrast, and `estimat`ing a population contrast. Syntax is

° The word `estimate`.

° A label, enclosed in quotes.

° The name of the independent variable. If you have more than one, or especially if it is a two or higher way design, consult the SAS manual under `proc glm`.

° The coefficients of the contrast (or more generally, the linear combination). Here, the weights are $a_1 = 1/2$, $a_2 = 1/2$, $a_3 = -1/2$ and $a_4 = -1/2$. We're estimating the difference between the average sales for 3-colour and 5-colour packages. The `divisor` option does not affect the tests. I did it so that `estimate` would really produce an estimate of the difference between average population means. The weights could have been decimal fractions, like

```
estimate '3Colourvs5Colour' package .5 .5 -.5  -.5;
```

but sometimes it is more convenient to enter integer coefficients (for example, if the denominator is 3). The output

```
                         General Linear Models Procedure

Dependent Variable: SALES   Number of Cases Sold

                                          T for H0:    Pr > |T|   Std Error of
Parameter                      Estimate    Parameter=0               Estimate

3Colourvs5Colour           -9.35000000          -6.25     0.0001     1.49705266
```

includes a t-test for whether the specified linear combination is different from zero. To treat this as a Scheffé test, calculate $F = t^2 = (-6.25)^2 = 39.0625$, so that $F_{sch} = F/3 = 13.02$. This is far greater than the critical value of 3.29, so we can conclude that (averaging over cartoon versus no-cartoon designs) sales for 5-colour packages are better than sales for 3-colour packages. Later, other evidence will cause us to question whether we should average over cartoon versus no cartoon, but for now the conclusion seems okay.

We can do this oneway anova with multiple regression. First consider indicator dummy variable coding with an intercept. Here is the part of the data step that defines the dummy variables. Because we have an intercept, we'll represent the four categories of package design with three dummy variables. The table below shows the model for population mean sales.

Design	p1	p2	p3	$E[Y] = \beta_0 + \beta_1 p1 + \beta_2 p2 + \beta_3 p3$
3Colour Cartoon	1	0	0	$\beta_0 + \beta_1 \quad = \mu_1$
3Col No Cartoon	0	1	0	$\beta_0 + \beta_2 \quad = \mu_2$
5Colour Cartoon	0	0	1	$\beta_0 + \beta_3 \quad = \mu_3$
5Col No Cartoon	0	0	0	$\beta_0 \quad\quad = \mu_4$

To clarify the parallel between population parameters and sample statistics, the corresponding table of estimated sales figures is

Design	p1	p2	p3	$\hat{Y} = b_0 + b_1 p1 + b_2 p2 + b_3 p3$
3Colour Cartoon	1	0	0	$b_0 + b_1 \quad = \overline{Y}_1$
3Col No Cartoon	0	1	0	$b_0 + b_2 \quad = \overline{Y}_2$
5Colour Cartoon	0	0	1	$b_0 + b_3 \quad = \overline{Y}_3$
5Col No Cartoon	0	0	0	$b_0 \quad\quad = \overline{Y}_4$

One thing these tables show is something that is true of *any* valid dummy variable coding scheme for a single categorical independent variable: $\hat{Y} = \overline{Y}$ for each category.

It is also easy to see that to test for differences among means, we want to simultaneously test whether β_1, β_2 and β_3 are different from zero -- or equivalently, whether b_1, b_2 and b_3 are *significantly* different from zero. That is, we want to simultaneosly test the dummy variables p1, p2 and p3. The overall F test of proc reg does the job.

```
proc reg;
     model sales = p1 p2 p3;
```

```
Model: MODEL1
Dependent Variable: SALES      Number of Cases Sold

                             Analysis of Variance

                                Sum of         Mean
       Source          DF      Squares       Square      F Value       Prob>F

       Model            3    588.22105    196.07368       18.591       0.0001
       Error           15    158.20000     10.54667
       C Total         18    746.42105
```

Wegot this same same F valuefor differences among the four means from `proc glm`. The next line does the 3 versus 5 color comparison.

```
      ncolour: test p1+p2 = p3; /* 3 vs 5 colours */
```

It works because we want to test whether $\frac{1}{2}(\mu_1 + \mu_2) = \frac{1}{2}(\mu_3 + \mu_4)$, and

$$\frac{1}{2}(\beta_0 + \beta_1 + \beta_0 + \beta_2) = \frac{1}{2}(\beta_0 + \beta_3 + \beta_0)$$

is agebraically equivalent to

$$\beta_1 + \beta_2 = \beta_3.$$

The estimate statement from `proc glm` yielded t = -6.25. Calculate F = t^2 = 39.0625, and compare the output of the `test` statement above:

```
Test: NCOLOUR  Numerator:    411.4000  DF:    1   F value:  39.0076
               Denominator:  10.54667  DF:   15   Prob>F:    0.0001
```

The difference is rounding error. It's the same test. But we'd rather avoid having to do algebra whenever we want to test a contrast. In cell means coding, we use an indicator dummy variable for each category (four, in this case), and omit the intercept. The tables that follow indicate why it's called cell mean coding.

Cell Means Coding for Package Design

Design	p1	p2	p3	p4	$E[Y] = \beta_1 p1 + \beta_2 p2 + \beta_3 p3 + \beta_4 p4$
3Colour Cartoon	1	0	0	0	$\beta_1 = \mu_1$
3Col No Cartoon	0	1	0	0	$\beta_2 = \mu_2$
5Colour Cartoon	0	0	1	0	$\beta_3 = \mu_3$
5Col No Cartoon	0	0	0	1	$\beta_4 = \mu_4$

Design	p1	p2	p3	p4	$\hat{Y} = b_0 + b_1 p1 + b_2 p2 + b_3 p3 + b_4 p4$
3Colour Cartoon	1	0	0	0	$b_1 = \overline{Y}_1$
3Col No Cartoon	0	1	0	0	$b_2 = \overline{Y}_2$
5Colour Cartoon	0	0	1	0	$b_3 = \overline{Y}_3$
5Col No Cartoon	0	0	0	1	$b_4 = \overline{Y}_4$

Here is the `proc reg.`

```
proc reg;
     model sales = p1 p2 p3 p4 / noint;
     alleq:    test p1=p2=p3=p4;
     numcol:   test p1+p2 = p3+p4;
     cartoon:  test p1+p3 = p2+p4;
     inter1:   test p1-p2 = p3-p4; /* Effect of cartoon depends on ncolours */
     inter2:   test p1-p3 = p2-p4; /* Effect of ncolours depends on cartoon */
     Y3_N3:    test p1=p2;  /* All pairwise tests */
     Y3_Y5:    test p1=p3;
     Y3_N5:    test p1=p4;
     N3_Y5:    test p2=p3;
     N3_N5:    test p2=p4;
     Y5_N5:    test p3=p4;
```

And the output. First, the overall F test, which is very different from what we had before.

```
Model: MODEL1
NOTE: No intercept in model. R-square is redefined.
Dependent Variable: SALES      Number of Cases Sold

                             Analysis of Variance

                                Sum of         Mean
       Source          DF      Squares        Square      F Value       Prob>F

       Model            4   7183.80000   1795.95000      170.286       0.0001
       Error           15    158.20000     10.54667
       U Total         19   7342.00000

           Root MSE       3.24756     R-square       0.9785
           Dep Mean      18.63158     Adj R-sq       0.9727
           C.V.          17.43042
```

With no intercept,

• • Total sum of squares is now $\sum_{i=1}^{n} Y_i^2$. It's no longer corrected for the mean; U means uncorrected. R^2 is radically affected

• • The overall F-test is for whether ALL the betas are zero - usually uninteresting

Notice now the parameter estimates are exactly the cell means.

```
                              Parameter Estimates

                      Parameter      Standard    T for H0:
     Variable  DF      Estimate         Error   Parameter=0    Prob > |T|

     P1         1     14.600000    1.45235441        10.053        0.0001
     P2         1     13.400000    1.45235441         9.226        0.0001
     P3         1     19.500000    1.62378159        12.009        0.0001
     P4         1     27.200000    1.45235441        18.728        0.0001
```

Now the custom tests:

```
Dependent Variable: SALES
Test: ALLEQ    Numerator:    196.0737  DF:    3   F value:  18.5911
               Denominator:  10.54667  DF:   15   Prob>F:    0.0001


Dependent Variable: SALES
Test: NUMCOL   Numerator:    411.4000  DF:    1   F value:  39.0076
               Denominator:  10.54667  DF:   15   Prob>F:    0.0001


Dependent Variable: SALES
Test: CARTOON  Numerator:     49.7059  DF:    1   F value:   4.7129
               Denominator:  10.54667  DF:   15   Prob>F:    0.0464

Dependent Variable: SALES
Test: INTER1   Numerator:     93.1882  DF:    1   F value:   8.8358
               Denominator:  10.54667  DF:   15   Prob>F:    0.0095


Dependent Variable: SALES
Test: INTER2   Numerator:     93.1882  DF:    1   F value:   8.8358
               Denominator:  10.54667  DF:   15   Prob>F:    0.0095


Dependent Variable: SALES
Test: Y3_N3    Numerator:      3.6000  DF:    1   F value:   0.3413
               Denominator:  10.54667  DF:   15   Prob>F:    0.5677


Dependent Variable: SALES
Test: Y3_Y5    Numerator:     53.3556  DF:    1   F value:   5.0590
               Denominator:  10.54667  DF:   15   Prob>F:    0.0399


Dependent Variable: SALES
Test: Y3_N5    Numerator:    396.9000  DF:    1   F value:  37.6327
               Denominator:  10.54667  DF:   15   Prob>F:    0.0001


Dependent Variable: SALES
Test: N3_Y5    Numerator:     82.6889  DF:    1   F value:   7.8403
               Denominator:  10.54667  DF:   15   Prob>F:    0.0135


Dependent Variable: SALES
Test: N3_N5    Numerator:    476.1000  DF:    1   F value:  45.1422
               Denominator:  10.54667  DF:   15   Prob>F:    0.0001


Dependent Variable: SALES
Test: Y5_N5    Numerator:    131.7556  DF:    1   F value:  12.4926
               Denominator:  10.54667  DF:   15   Prob>F:    0.0030
```