Chapter 3

Comparing Several Means

3.1 One-way analysis of variance

This chapter starts with the humble one-way (one-factor) analysis of variance (ANOVA). It is called *one* way because there is a single categorical independent variable. This categorical independent variable, which may be either observed or experimentally manipulated, divides the sample into *groups* of observations. The objective is to test for differences among means. Note that because the independent variable divides the cases into groups, it is a between-subjects factor. Within-subjects (repeated measures) techniques will be discussed later.

Assumptions The test assumes independent random sampling from each sub-population, and also that the dependent variable has a conditional distribution that is normal, with equal variances. That is, for each value of the categorical independent variable, there is a sub-population (perhaps hypothetical), and the dependent variable is normally distributed within that sub-population. While the population means of all the normal distributions may differ, their population variances are all identical.

A normal distribution is completely specified by its mean and variance, and we are assuming that the variances are all equal. So if the means of the conditional distributions are also equal, then the conditional distributions are identical. This makes the independent and dependent variable *unrelated* by the definition in Chapter 1. Thus we see that in the one-way ANOVA, the only possible kind of population relationship between the independent variable and the dependent variable is a difference among group means.

The "assumptions" of a statistical test actually represent a mathematical *model* for the data, and that model is used to formally derive the test. Such derivations are always hidden in applied classes. But it makes a practical difference, because some assumptions are often violated in practice, and frequently these assumptions were adopted in the first place to make the model mathematically tractable, not because anybody seriously believed they would be true for the typical data set.

Sometimes, the assumptions that allow the mathematical derivation of a test are not really necessary. The test might work, or anyway work pretty well, even if the assumptions are violated. When this is the case, the test is said to be *robust* with respect to those assumptions. Usually, robustness is something that starts to happen as the sample size gets large, if it happens at all.

When we say a test "works," we mean two things

- It protects against Type I error (false significance) at something close to the stated level. That is, if nothing is really going on, significant effects will be falsely detected at the 0.05 level not much more than 5% of the time.
- The power of the test is reasonably good. At the very least, power (the probability of correctly rejecting the null hypothesis) increases when the relationship between independent variable and dependent variable becomes stronger, and also increases with the sample size, approaching one as the sample size approaches infinity for *any* non-zero relationship between variables.

For the one-way analysis of variance (and for factorial[1] ANOVA in general) if the assumption of equal variances holds but the normal assumption does not, the test is robust for large samples. The rough rule would be $n = 20$ to 25 for each group, though for data that are sufficiently non-normal, an arbitrarily large sample might be required. If the equal variances assumption is violated, then the test is robust for large samples if the sample sizes for each group are approximately equal. Here, the meaning of "large" is murky.

***Analysis* of variance** The word *analysis* means to take apart or split up, and in the analysis of variance, variation in the dependent variable is split into two components: variation of the data values that is explained by the independent variable (Sum of Squares Between groups), and variation that is left unexplained (Sum of Squares Within groups). Here's how it goes.

Suppose we want to predict the value of a dependent variable, without using any independent variables yet. The best prediction (in the sense of least squares) is the sample mean. Subtract the sample mean from each dependent variable value, and we obtain a set of *deviations* representing errors of prediction. Squaring these deviations to remove the sign and adding them up yields a measure of the total variation in the sample data. We call it the Total Sum of Squares, or $SSTO$.

The total sum of squares is the total amount of variation in the dependent variable. It is what any potential predictor would seek to explain. Here, the word "explain" really means "reduce." To the extent that the total squared error of prediction around a predictor is *less* than $SSTO$, the predictor is effective. It has "explained" part of the variation in the dependent variable — at least in the sense of taking care of it.

Now consider a categorical independent variable as a predictor of the dependent variable. This variable (which could be either an experimental treatment or an existing variable that is merely assessed, like breed of dog) subdivides the cases into two or more

[1] The term "factor" is another term for categorical independent variable. Factorial research designs imply analyses with one or more categorical independent variables, usually more than one.

groups. Now, if you want to predict the dependent variable, you would use the *group* mean rather than the overall mean. For example, if you want to predict the amount of food eaten by an Irish wolfhound, you would use the mean consumption of the Irish wolfhounds in your sample, not the mean consumption of all the dogs combined.

No matter how good a predictor is, it will not be perfect for real data. For each value of the dependent variable, subtract off the group mean (not the overall mean, this time). Square those errors of prediction, add them up, and we have the Sum of Squared error of prediction Within groups, where the dependent variable is being predicted from group membership. The initials SSW stand for Sum of Squares Within. This quantity represents the variation in the dependent variable that is *not* explained by the independent variable. It is left over, or *residual.*[2]

If $SSTO$ is the total amount of variation that could be explained, and SSW is the amount of variation that is left unexplained, then the difference between them must be the variation that is explained. Now suppose that by some amazing coincidence, all the group means were exactly equal. Then $SSW = SSTO$, and absolutely no variation is explained by the independent variable. This suggests that explained variation must be linked to variation between group means, and we write

$$SSTO = SSB + SSW,$$

where SSB, which stands for "Sum of Squares Between," is the variation that is explained by the categorical independent variable.

The notation SSB for the explained sum of squares is supported by a set of formulas, which are given because they may be illuminating for some readers, not because you will ever have to use them for calculation. First, suppose that there are p groups,[3] with n_j cases in each group, $j = 1, \ldots, p$. The total sample size is $n = \sum_{j=1}^{p} n_j$. Observation i in group j is denoted by $Y_{i,j}$, and the sample means are

$$\overline{Y}_j = \frac{\sum_{i=1}^{n_j} Y_{i,j}}{n_j} \text{ and } \overline{Y} = \frac{\sum_{j=1}^{p} \sum_{i=1}^{n_j} Y_{i,j}}{n}.$$

[2] The differences between the data values and group means are *residuals.* In regression, the predictions are points on the regression line or surface, and again the residuals are differences between observed and predicted values. In regression, the initials SSE stand for Sum of Squared Error of prediction. SSW is a special kind of SSE.

[3] This p is different from the p-value. It connects so well with standard notation in multiple regression that we're going to use it for the number of groups, even though it's unfortunate when the same symbol is used for two different things. You'll just have to realize which p is meant from the context.

Then, the formulas for the sums of squares are

$$\begin{aligned}
SSB &= \sum_{j=1}^{p} n_j(\overline{Y}_j - \overline{Y})^2 \\
SSW &= \sum_{j=1}^{p}\sum_{i=1}^{n_j}(Y_{i,j} - \overline{Y}_j)^2 \\
SSTO &= \sum_{j=1}^{p}\sum_{i=1}^{n_j}(Y_{i,j} - \overline{Y})^2.
\end{aligned}$$

You can see that the Sum of Squares Between groups is literally the variation of the group means around the overall mean, with the contribution of each squared deviation determined by the group sample size. Again, the sums of squares add up: $SSTO = SSB + SSW$.

ANOVA summary tables Sums of squares and related quantities are often presented in an *Analysis of variance summary table.* In the old days, these were given in the results sections of journal articles; today, they appear only in the output printed by statistics packages. There are minor differences in detail. SAS `proc glm` produces one in this format.

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	$p-1$	SSB	$MSB = SSB/(k-1)$	MSB/MSW	p-value
Error	$n-p$	SSW	$MSW = SSW/(n-k)$		
Corrected Total	$n-1$	$SSTO$			

Sums of squares add up, degrees of freedom add up, Mean Square = SS/df, and F is the ratio of two Mean Squares. The F ratio is the test statistic for

$$H_0 : \mu_1 = \ldots = \mu_p.$$

That is, under the null hypothesis all the population means are equal.

For a particular data set, the analysis of variance summary table will be filled with numbers. It allows you to calculate a very useful descriptive statistic:

$$R^2 = \frac{SSB}{SSTO}.$$

R^2 is the **proportion of the variation in the dependent variable that is explained by the independent variable**.[4] This is exactly the interpretation we give to the square of the correlation coefficient; R^2 is a reasonable index of how strongly the dependent variable is related to the independent variable.

If the sample size is small, it is possible for R^2 to be fairly large, but the differences among means are not statistically significant. Or, if the sample size is huge, even a very weak, trivial relationship can be "significant." To take an extreme example, one fabled analysis of U. S. census data found virtually *everything* to be statistically significant, even average shoe size East versus West of the Mississippi River. You might say that there are really two kinds of significance: statistical significance and *substantive* significance. R^2 can help you assess substantive significance. Confidence intervals can be useful, too.

What's a good value of R^2? Traditions vary in different scientific disciplines. Not surprisingly, areas dominated by noisy data and weak relationships are more tolerant of small R^2 values. My personal preference is guided by the correlation coefficient. In a scatterplot, the correlation has to be around 0.30 in absolute value before I can really tell whether the relationship is positive or negative. Since $0.30^2 = 0.09$, I start taking independent variables seriously once they explain around nine or ten percent of the variation (or of the *remaining* variation, if there are multiple independent variables). But opinions differ. Cohen's (1988) authoritative *Statistical power analysis for the behavioral sciences* [6] suggests a much more modest standard.

3.2 Testing Contrasts

The F-test from a one-way ANOVA is useful, but it usually does not tell you all you need to know. For example, if the test is significant, the conclusion is that not all the group means are equal in the population. But you do not know which means are different from each other. Or, specific comparisons might be of interest. For example, you may have reason to believe that the response to drug A is better than the average response to drugs B, C and D. Fortunately, analysis of variance technology can do much more than simply test for equality of several group means. First, we need a few definitions.

A *linear combination* is a weighted sum of several quantities. It has the general form

$$\text{Linear Combination} = a_1Q_1 + a_2Q_2 + \ldots + a_kQ_p.$$

The symbols a_1 through a_p stand for numerical constants. We will call these the *weights* of the linear combination.

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. A population contrast has the form

$$c = a_1\mu_1 + a_2\mu_2 + \cdots + a_p\mu_p$$

[4] Psychologists often call it the proportion of *variance* that is explained, while statisticians usually call it proportion of sum of squares. The "proportion of variance" terminology can be justified in a couple of different ways, and is perfectly okay.

where $a_1 + a_2 + \cdots + a_p = 0$. The case where all of the a values are zero is uninteresting, and is excluded. A population contrast is estimated by a sample contrast:

$$\widehat{c} = a_1\overline{Y}_1 + a_2\overline{Y}_2 + \cdots + a_p\overline{Y}_p.$$

With the right software (and that definitely includes SAS), it is easy to test whether any contrast equals zero, and to obtain a confidence interval for a contrast. It is also easy to test several contrasts at once.

By setting $a_1 = 1$, $a_2 = -1$, and the rest of the a values to zero we get $L = \overline{Y}_1 - \overline{Y}_2$, so it's easy to see that any difference between two means is a contrast.[5] Also, the average of one set of means minus the average of another set is a contrast.

The F test for equality of p means can be viewed as a simultaneous test of $p-1$ contrasts. For example, suppose there are four treatments, and the null hypothesis of the initial test is $H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4$. The table gives the a_1, a_2, a_3, a_4 values for three contrasts; if all three contrasts equal zero then the four population means are equal, and *vice versa.*

a_1	a_2	a_3	a_4
1	-1	0	0
0	1	-1	0
0	0	1	-1

The way you read this table is

$$\begin{array}{ccccccccc}
\mu_1 & - & \mu_2 & & & & & = & 0 \\
 & & \mu_2 & - & \mu_3 & & & = & 0 \\
 & & & & \mu_3 & - & \mu_4 & = & 0
\end{array}$$

Clearly, if $\mu_1 = \mu_2$ and $\mu_2 = \mu_3$ and $\mu_3 = \mu_4$, then $\mu_1 = \mu_2 = \mu_3 = \mu_4$, and if $\mu_1 = \mu_2 = \mu_3 = \mu_4$, then $\mu_1 = \mu_2$ and $\mu_2 = \mu_3$ and $\mu_3 = \mu_4$. The simultaneous F test for the three contrasts is 100% equivalent to what you get from a one-factor ANOVA; it yields the same F statistic, the same degrees of freedom, and the same p-value.

There is always more than one way to set up the contrasts to test a given hypothesis. Staying with the example of testing differences among four means, we could have specified

a_1	a_2	a_3	a_4
1	0	0	-1
0	1	0	-1
0	0	1	-1

so that all the means are equal to the last one,[6] and thus equal to each other. No matter how you set up collection of contrasts, if you do it correctly you always get the same test statistic and p-value.

[5] The test of a contrast between two means is not exactly the same as what you would get if you ignored all the data from the other groups, and just did a two-sample t-test or a one-way analysis with two groups. This is because the test of a contrast uses data from *all* the groups to estimate the common within-group variance (it uses Mean Squared Within from the full one-way ANOVA).

[6] These contrasts (differences between means) are actually *equal* to the regression coefficients in a multiple regression with indicator dummy variables, in which the last category is the reference category. More on this later.

3.3 The Tubes Data

In the *tubes data* (kindly provided by Linda Kohn of the University of Toronto's Botany department), the investigators were studying sclerotial fungi. The fungus they were studying is nasty black stuff that looks much like the fungus that grows between the tiles above your bathtub (well, okay, my bathtub). The fungus is called "sclerotial" because that is how they reproduce. Sclerotia are little pods that produce spores. When the pod opens and the spores are released, they float through the air, land somewhere, and maybe start to grow.

Ordinarily, these sclerotial fungi grow on plants. In fact, they often grow on canola plants, and kill them or impair their growth. The canola plant produces a high-quality vegetable oil, and is one of Canada's biggest cash crops. So this makes a difference, because it is about food.

All these fungi look the same, but they are not. There are different strains of fungus, and the investigators know how to do genetic fingerprinting to tell them apart. The different types are called "mycelial compatibility groups" (MCG for short), because if you grow two different genetic types together in a dish, they will separate into two visibly distinct colonies, and stay separated. The stuff that grows together is compatible. Before techniques of genetic fingerprinting were developed, this was the only way to tell the strains of apart.

The MCGs are genetically and spatially distinct, but do some grow faster than others? This could have implications for agricultural practice as well as science. In this experiment, the fungus is not growing on plants; it's growing in "race tubes," in a nutrient solution. The implicit assumption here is that types of fungus that grow better in test tubes will also grow better on plants. Is this true? It's definitely an empirical question, because plants fight off these infestations with something like an immune system response, and the fungus that grows best on a completely passive host is not necessarily the one that will grow best on a host that is fighting back. This is an issue of external validity; see Section 1.3.

There are six MCGs, with four test tubes each. So, there are $n = 24$ cases in all. This may seem like a very small sample size, and in fact the sample size was not chosen by a power analysis (see Section 1.2.1 in Chapter 1 for a brief discussion) or any other systematic method. It was entirely intuitive — but this is the intuition of scientists with well-deserved international reputations in their field. Here's how they thought about it.

The samples of each fungus type are genetically identical, the test tubes in which they are placed are exactly identical, and the nutrient solution in the tubes comes from one well-mixed batch; it's exactly the same in all tubes. The amount of nutrient solution in each tube is placed by hand, but it's done *very* carefully, by highly trained and experienced personnel. The temperature and humidity of the tubes in the lab are also carefully controlled, so they are the same, except for microscopic differences. Really, the only possible source of variation in measured growth (except for very tiny errors of measurement) is the genetic makeup of the fungus. Under the circumstance, one tube for each fungus type might seem adequate to a biologist (though you couldn't do any significance tests), two tubes would be replicating the study, and four tubes per condition might seem like overkill.[7] We will see presently that this intuition is supported by how the statistical analysis turned out.

Every day for two weeks, a lab assistant (maybe a graduate student) measured each tube, once in the morning and once in the evening. She measured the length of fungus in centimeters, and also counted the sclerotia, as well as taking other measurements. We will confine ourselves to a single dependent variable – length of the fungus on the evening of day 10. After that point, the fastest-growing strains spread past the end of the test tubes, creating a pattern of missing data that is too challenging to be considered here. So, we have fungus type, a categorical independent variable called `MCG` that takes on six values (the codes are numerical, and they are informative to the botanists); and we have the single dependent variable `pmlng10`, which roughly indicates growth rate.

The The SAS program `tubes09f.sas` contains a one-way analysis of variance with many (not all) of the bells and whistles. The strategy will to present the complete SAS program first and then go over it piece by piece and explain what is going on – with one major statistical digression. Here is the program.

```
/*************** tubes09f.sas ****************/
/*      One-way analysis of tubes data       */
/*********************************************/

%include 'tuberead2.sas';
title2 'One-way analysis of tubes data';

proc freq;
     tables mcg;

proc glm;
     title3 'Just the defaults';
     class mcg;
     model pmlng10 = mcg;

/* For convenience, MCGs are:  198 205 213 221 223 225  */

proc glm;
```

[7] It is true that with this small sample, the assumptions of normal distribution and equal variance are basically uncheckable. But they can be justified as follows. The only reason that the length measurement for a particular type of fungus would not be completely identical would be a multitude of tiny, more or less independent random shocks arising from tiny errors of measurement (the lab assistant is using a ruler) and even smaller differences in the chemical composition of the nutrient solution and micro-climate within the growth chamber. These random shocks may not be identically distributed, but as long as they are independent and fairly numerous, a version of the Central Limit Theorem assures us that their sum is normally distributed. Also, since code numbers were used to label the test tubes (the lab assistants were blind to experimental condition), there is no reason to expect that the nature of the random shocks would differ for the different fungus types. This justifies the assumption of equal variances.

```
     title3 'With contrasts and multiple comparisons';
     class mcg;
     model pmlng10 = mcg / clparm; /* clparm give CI for contrasts down in
                                      the estimate statement. */
     means mcg;
     /* Test custom contrasts, or "planned comparisons" */
     contrast '198vs205'       mcg   1   -1     0   0   0   0;
     contrast "223vs225"       mcg   0    0     0   0   1  -1;
     contrast  '223n225vsRest' mcg  -1   -1    -1  -1   2   2;
     /* Test equality of mcgs excluding 198: a COLLECTION of contrasts */
     contrast 'AllBut198'      mcg   0   1  -1   0   0   0,
                               mcg   0   0   1  -1   0   0,
                               mcg   0   0   0   1  -1   0,
                               mcg   0   0   0   0   1  -1;
     /* Replicate overall F test just to check. */
     contrast 'OverallF=76.70' mcg   1  -1   0   0   0   0,
                               mcg   0   1  -1   0   0   0,
                               mcg   0   0   1  -1   0   0,
                               mcg   0   0   0   1  -1   0,
                               mcg   0   0   0   0   1  -1;
     /* Estimate will print the value of a sample contrast and do a t-test
        of H0: Contrast = 0 */
     /* F = t-squared */
     estimate '223n225vsRest'  mcg -.25 -.25 -.25 -.25 .5 .5;
     estimate 'AnotherWay'     mcg  -3   -3    -3  -3   6  6 / divisor=12;
     /* Multiple Comparisons */
     means mcg / Tukey Bon Scheffe; /* Simultaneous Confidence Intervals */
     /* Tables of adjusted p-values -- more convenient */
     lsmeans mcg / pdiff adjust=bon;
     lsmeans mcg / pdiff adjust=tukey;
     lsmeans mcg / pdiff adjust=scheffe;

/* Get Scheffe critical value from proc iml */

proc iml;
     title2 'Scheffe critical value for all possible contrasts';
     numdf = 5;   /* Numerator degrees of freedom for initial test */
     dendf = 17;  /* Denominator degrees of freedom for initial test */
     alpha = 0.05;
     critval = finv(1-alpha,numdf,dendf);
     scrit = critval * numdf;

     print "Initial test has"  numdf " and " dendf "degrees of freedom."
           "----------------------------------------------------------"
           "Using significance level alpha = " alpha
           "------------------------------------------------"
           "Critical value for the initial test is " critval
           "------------------------------------------------"
           "Critical value for Scheffe tests is " scrit
           "------------------------------------------------";
```

The program begins with `%include 'tuberead2.sas';` the data step is contained in a separate file called `tuberead2.sas`, not shown here. The `%include` statement reads in the external file. This is what was done with the `statclass` data presented in Section 2.2.5 of Chapter 2. More detail about `%include` is given there.

Then (after the second title line) we request a frequency distribution of the independent variable – always a good idea.

```
proc freq;
     tables mcg;
```

Here is the output of `proc freq`.

```
       Fungus Tube data with line1=113 eliminated                  1
             One-way analysis of tubes data

                   The FREQ Procedure

              Mycelial Compatibility Group

                                Cumulative    Cumulative
mcg    Frequency     Percent     Frequency      Percent
--------------------------------------------------------
198           4       17.39             4        17.39
205           4       17.39             8        34.78
213           3       13.04            11        47.83
221           4       17.39            15        65.22
223           4       17.39            19        82.61
225           4       17.39            23       100.00
```

The first line of the title contains a reminder that one of the cases (tubes) has been eliminated from the data. In the full data set, there was an outlier; when the biologists saw it, they were absolutely convinced that in spite of the great care taken in the laboratory, the tube in question had been contaminated with the wrong strain of fungus. So we set it aside. This is why there are only three test tubes in the `mcg=213`, group, and four in all the others.

Next, we have a bare-bones `proc glm`. The initials stand for “General Linear Model,” and indeed the procedure is very general. Especially in this first example, we are just scratching the surface. All the parts are obligatory except `title3`, which produces a third title line that is displayed only for the output of this procedure.

```
proc glm;
     title3 'Just the defaults';
     class mcg;
     model pmlng10 = mcg;
```

The `class` statement declares package to be categorical. Without it, `proc glm` would do a regression with `mcg` as a quantitative independent variable. The syntax of the minimal `model` statement is

`model` Dependent variable(s) `=` Independent variable(s)`;`

Here is the output; it's part of the list file.

```
--------------------------------------------------------------------------------

                  Fungus Tube data with line1=113 eliminated                   2
                        One-way analysis of tubes data
                               Just the defaults

                               The GLM Procedure

                            Class Level Information

               Class         Levels    Values

               mcg                6    198 205 213 221 223 225


                    Number of Observations Read          23
                    Number of Observations Used          23

--------------------------------------------------------------------------------

                  Fungus Tube data with line1=113 eliminated                   3
                        One-way analysis of tubes data
                               Just the defaults

                               The GLM Procedure

Dependent Variable: pmlng10
```

```
                                        Sum of
Source                      DF         Squares     Mean Square    F Value    Pr > F

Model                        5     55.43902174     11.08780435      76.70    <.0001

Error                       17      2.45750000      0.14455882

Corrected Total             22     57.89652174


             R-Square     Coeff Var      Root MSE    pmlng10 Mean

             0.957554      1.500224      0.380209        25.34348



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

mcg                          5     55.43902174     11.08780435      76.70    <.0001


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

mcg                          5     55.43902174     11.08780435      76.70    <.0001
```

First, `proc glm` gives "Class Level Information: " the name of the independent variable, the number of "Levels" (groups), and the actual values taken on by the independent variable. Then we get the sample size ($n = 23$). That's all for Page 2 of the output. If not for the `formdlim` option, SAS would print the next page of output on a new physical sheet of paper.

On the next page of output (that is, the next *logical* page, as opposed to physical page), SAS first prints the title lines, then the name of the dependent variable, and the first of three analysis of variance summary tables. It's a standard one, and leads to the F value of 76.70; this is the "numerical value of the test statistic (so often requested in homework problems) for testing equality of means. The p-value is tiny ($p < 0.0001$). The differences among means are statistically significant, but with this minimal output we cannot even guess which means might be significantly different from which others; the sample means are not even displayed.

On the other hand, we do get some other statistics. Reading from right to left, we see the sample mean of the dependent variable, `Root MSE` (literally the square root of the Mean Square Within groups), The Coefficient of Variation (100 times `Root MSE` divided by $\overline{Y}$, for what that's worth), and

$$R^2 = \frac{SSB}{SSTO} = \frac{55.4390}{57.8965} = 0.957554.$$

That is, nearly 96% of the variation in growth rate is explained by genetic the type of the fungus. This is an overwhelmingly strong relationship between the independent and dependent variables, and completely justifies the investigators' judgement that a small sample was all they needed. You'd never see anything this strong outside the laboratory (say, in a canola field).

Next in the SAS program comes the *real* `proc glm` — one that illustrates testing and confidence intervals for contrasts, and also multiple comparisons (sometimes called *post hoc* tests, or *probing*). It starts like the one we've just examined.

```
/* For convenience, MCGs are:  198 205 213 221 223 225  */

proc glm;
     title3 'With contrasts and multiple comparisons';
     class mcg;
     model pmlng10 = mcg / clparm; /* clparm give CI for contrasts down in
                                      the estimate statement. */
     means mcg;
```

The comment lists the `mcg`s (values of the independent variable) in order; it's useful here for setting up contrasts and remembering what they mean. This `proc glm` starts out just like the last one, except for the `clparm` option on the `model` statement; `clparm` stands for "confidence limits for parameters." The parameters in question are contrasts (which are actually *functions* of several model parameters), requested later in the `estimate` statements. This is the best way to obtain confidence intervals for contrasts.

There's also an optional means statement that goes `means mcg`. It requests a display of the sample means of the dependent variable, separately for each value of the independent variable named. A `means` statement is really necessary in any oneway ANOVA with `proc glm` if you are to have any idea of what is going on. But the SAS *syntax* does not require it. Here is the table of means generated by the means statement.

```
                          The GLM Procedure

               Level of           -----------pmlng10-----------
               mcg          N             Mean          Std Dev

               198          4       28.3250000       0.35939764
               205          4       25.8500000       0.28867513
               213          3       25.0000000       0.26457513
               221          4       23.4000000       0.48304589
               223          4       24.8000000       0.16329932
               225          4       24.6000000       0.54772256
```

Next, we request test of some contrasts, and also tests of two *collections* of contrasts. As the comment in the program indicates, these are sometimes called “planned comparisons” of treatment means. The implication is that they are tests of specific hypotheses that were developed before looking at the data – maybe the hypotheses that the study was designed to test in the first place. Maybe.

```
/* Test custom contrasts, or "planned comparisons" */
contrast ’198vs205’       mcg   1   -1     0   0   0   0;
contrast "223vs225"       mcg   0    0     0   0   1  -1;
contrast ’223n225vsRest’  mcg  -1   -1    -1  -1   2   2;
/* Test equality of mcgs excluding 198: a COLLECTION of contrasts */
contrast ’AllBut198’      mcg   0   1  -1   0   0   0,
                          mcg   0   0   1  -1   0   0,
                          mcg   0   0   0   1  -1   0,
                          mcg   0   0   0   0   1  -1;
/* Replicate overall F test just to check. */
contrast ’OverallF=76.70’ mcg   1  -1   0   0   0   0,
                          mcg   0   1  -1   0   0   0,
                          mcg   0   0   1  -1   0   0,
                          mcg   0   0   0   1  -1   0,
                          mcg   0   0   0   0   1  -1;
```

The syntax of the `contrast` statement is (reading left to right):

1. The word `contrast`

2. A label for the contrast (or set of contrasts), enclosed in single or double quotation marks

3. The name of the categorical independent variable. If there is more than one categorical independent variable (factor), you’ll get a contrast of the *marginal* means averaging across the other factors.

4. The weights of the contrast — the constants $a_1, \ldots, a_p$ described in Section 3.2.

5. If you want to test more than one contrast simultaneously, separate the contrasts by commas, as in the example. You must repeat the name of the categorical independent variable each time.

6. End the statement with a semicolon, as usual.

If the weights $a_1, \ldots, a_p$ do not add up to zero, you won’t get a test of whether the resulting linear combination equals zero. You don’t even get an error message or warning, just a ”Note” on the log file saying something like “CONTRAST LC is not estimable.” This actually makes perfectly good sense if you understand the way that `proc glm` parameterizes linear models that have categorical independent variables. But the waters are a bit deep here, so we’ll let it go for now.

The output of the contrast statement comes after the ANOVA summary table and after the output of the means statement (and `lsmeans`), even if you request means after you've requested contrasts. They are nicely labelled, using the labels supplied in the `contrast` statements. Naturally, the overall F value of 76.70 appearing in the label of the last test was obtained in an earlier run.

```
                                 The GLM Procedure

Dependent Variable: pmlng10

Contrast                     DF    Contrast SS    Mean Square   F Value   Pr > F

198vs205                      1    12.25125000    12.25125000     84.75   <.0001
223vs225                      1     0.08000000     0.08000000      0.55   0.4671
223n225vsRest                 1     4.62182432     4.62182432     31.97   <.0001
AllBut198                     4    12.39526316     3.09881579     21.44   <.0001
OverallF=76.70                5    55.43902174    11.08780435     76.70   <.0001
```

Next we have the `estimate` statement, which has a syntax similar to `contrast`. It is limited to single contrasts. They have to be actual contrasts, and not just generic linear combinations of cell means. The `estimate` statement prints the value of the sample contrast, a number that is an *estimate* of the population contrast. You also get a two-sided t-test of the null hypothesis that the contrast equals zero in the population. This is equivalent to the F-test generated by `contrast`; $F = t^2$, and the p-values are identical.

Notice that if you are just interested in a test for whether a contrast equals zero, multiplying by a constant has no effect – so the test of $-0.5, -0.5, 1.0$ is the same as the test for $1, 1, -2$; you'd probably use `contrast`. But if you are using `estimate`, you probably are interested in the numerical value of the contrast, often the difference between two means or averages of means. Some of these can be awkward to specify in decimal form, so you can use integers and give a divisor, as shown below.

```
    /* Estimate will print the value of a sample contrast and do a t-test
       of H0: Contrast = 0 */
    /* F = t-squared */
    estimate '223n225vsRest'  mcg -.25 -.25 -.25 -.25 .5 .5;
    estimate 'AnotherWay'     mcg  -3   -3    -3  -3   6  6 / divisor=12;
```

Here is the output of `estimate`. As mentioned earlier, the confidence limits were produced by the `clparm` option on the `model` statement.

```
                                            Standard
Parameter                   Estimate           Error    t Value    Pr > |t|

223n225vsRest            -0.94375000      0.16690623      -5.65      <.0001
AnotherWay               -0.94375000      0.16690623      -5.65      <.0001

             Parameter                95% Confidence Limits

             223n225vsRest            -1.29589137  -0.59160863
             AnotherWay               -1.29589137  -0.59160863
```

3.4 Multiple comparisons

The `means` statement of `proc glm` lets you look at the group means, but it does not tell you which means are significantly different from which other means. 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, so we can disregard "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 well be about 0.05, depending on how well the assumptions of the test are met. But suppose you do a *lot* of tests on a data set that is purely noise, with no true relationships between any independent variable and any dependent variable. Then 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. In the tubes data, there are 6 different fungus types, and thus 15 potential pairwise comparisons.

You have to pity the poor scientist[8] who learns about this and is honest enough to take the problem seriously. 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 several ways to resolve the problem. One way is to basically ignore it, 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

[8] Let's use the term "scientist" generously to apply to anyone trying to obtain informmation from a set of numerical data.

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.

My favourite solution is to collect enough data so that they can be randomly split into an exploratory and a replication sample. You explore one of the samples thoroughly, doing all sorts of tests, maybe re-defining the variables in the process. The result is a set of very specific hypotheses. Then you test the hypotheses on the second data set. This is great, unless the data are very time-consuming or expensive to collect. In that case, you're lucky to have one small data set, and you have to use all of it at once or you won't have enough power to detect anything.

Taking this unfortunate reality into account, statisticians have looked 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. For now, let's concentrate on following up a significant F test in a one-way analysis of variance. Three approaches will be presented, named after their originators: Bonferroni[9], Tukey and Scheffé. There are many more.

3.4.1 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

[9] Actually, Mr. Bonferroni is only indirectly responsible for the Bonferroni method of multiple comparisons. He gets credit for the probability inequality that says $P(\cup_{j=1}^k A_j) \leq \sum_{j=1}^k P(A_j)$. Letting A_j be the event that null hypothesis j is rejected (assume they are all true), we get the Bonferroni multiple comparison method quite easily.

correction, the difference in reported liking between worms and spinach soufflé 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 with a Bonferroni correction is less than or equal to 0.05 – almost always 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 at the level you want[10].

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.

3.4.2 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 Scheff can sometimes be more powerful.

[10] 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, without any correction for multiple testing. 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? This will be illustrated using the Math data described at the end of Chapter 2

3.4.3 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. 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 followups to a one-way ANOVA, SAS tests all pairwise differences between means, but *there are infinitely many more contrasts in the same family 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 a license for unlimited data fishing, at least within 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 single 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 statistically significant). This is part of the default output from `proc glm` when you request Scheffé tests using the `means` statement – or you can use `proc iml`[11].

A contrast is significantly different from zero by a Scheffé test if the F statistic is greater than the usual critical value *multiplied by* $p-1$, where p is the number of groups. You can get the F statistics with `contrast`. Keep doing tests until you run out of ideas.

Notice that multiplying 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 Mr. 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. The power of a Scheffé test is the probability of getting an F bigger than the critical value *multiplied by* $p-1$. You can pay for it by increasing the sample size.

Which method should you use? In most practical data analysis situations, you would only use one of the three multiple comparison methods. Here are some guidelines.

[11] Or, you could even use a table of critical values in the back of a Statistics text book. The exact degrees of freedom you want probably won't be in there, so you'll have to interpolate. Yuk.

- If the sample sizes are nearly equal and you are only interested in pairwise comparisons, use Tukey because it's most powerful in this situation.

- If the sample sizes are not close to equal and you are only interested in pairwise comparisons, there is (amazingly, just this once) no harm in applying all three methods and picking the one that gives you the greatest number of significant results. This is because you *could* calculate the three types of adjusted critical value in advance before seeing the data, and choose the smallest one.

- If you are interested in testing contrasts that go beyond pairwise comparisons and you can specify *all* of them (exactly what they are, not just how many) before seeing the data, Bonferroni is almost always more powerful than Scheffé. Tukey is out, because it applies only to pairwise comparisons.

- If you want lots of special contrasts but you don't know exactly what they all are, Scheffé is the only honest way to go, unless you have a separate replication data set.

3.4.4 Simultaneous confidence intervals and adjusted p-values

The Bonferroni and Scheffé methods allow you to test an arbitrary family of contrasts simultaneously, while holding down the *joint* Type I error rate. If you want to test a contrast that is a little special or unusual, you'd use the test from the `contrast` or `estimate` statement, along with an adjusted critical value. But if you're only interested in comparing all possible pairs of group means, you don't have to specify all those contrasts; SAS does it for you. Two equivalent formats are available, simultaneous confidence intervals and adjusted p-values. *Equivalent* means that both methods label exactly the same differences as significant;the only difference is in how the results are printed.

Simultaneous confidence intervals When you invoke multiple comparisons using the `means` statement (this is the older way), as in

```
means package / Tukey Bon Scheffe;
```

you get our three favourite kinds of multiple comparisons for all pairwise differences among means. (SAS is not case sensitive, so capitalizing the names is not necessary.) The multiple comparisons are presented in the form of simultaneous confidence intervals. If the 95% confidence interval does not include zero, the test (Bonferroni, Tukey or Scheffé) is significant at the joint 0.05 level. The confidence intervals are correct, but they are ugly to look at and not recommended. No output from the command above will be shown.

Adjusted p-values Adjusted p-values are adjusted for the fact that you are doing multiple tests; you believe the results when the adjusted p-value is less than 0.05. The adjustment is easy to describe for the Bonferroni method; just multiply the ordinary p-value by the number of tests, and if the resulting value is more than one, call it 1.00. For the Scheffé method, divide the computed value of F by $p-1$; the Scheffé adjusted

p-value is the tail area of the F distribution above this value. I don't know exactly how the Tukey p-value adjustment works, but if you really need to know you can look it up in the SAS documentation.

While the `means` statement allows you to request several different multiple comparison methods at once, `lsmeans` must be invoked separately for each method you want. Here is the syntax.

```
lsmeans mcg / pdiff adjust=bon;
lsmeans mcg / pdiff adjust=tukey;
lsmeans mcg / pdiff adjust=scheffe;
```

The keyword `lsmeans` stands for "least squares means," which are the group means adjusted for one or more quantitative independent variables (covariates). Since there are no quantitative independent variables here, the least squares means are the same as ordinary means.[12]

The syntax of the `lsmeans` is (reading from left to right)

- `lsmeans`
- The name of the independent variable
- A slash; options are given to the right of the slash.
- `pdiff` requests a table of p-values for testing all pairwise differences between means.
- `adjust=` and the name of the method. Use "bon" or "Bon" instead of the full name.

Here is the Scheffé output. First we get the (least squares) means, and then a table showing the adjusted p-values. The number in row j, column k contains the adjusted p-value for the test of mean j against mean k.

```
                         The GLM Procedure
                        Least Squares Means
              Adjustment for Multiple Comparisons: Scheffe

                                pmlng10      LSMEAN
                   mcg           LSMEAN      Number

                   198       28.3250000           1
                   205       25.8500000           2
                   213       25.0000000           3
                   221       23.4000000           4
                   223       24.8000000           5
                   225       24.6000000           6
```

[12] Least squares means will be explained properly in a later chapter, using concepts from multiple regression.

```
                        Least Squares Means for effect mcg
                       Pr > |t| for H0: LSMean(i)=LSMean(j)

                           Dependent Variable: pmlng10

 i/j           1           2           3           4           5           6

    1                 <.0001      <.0001      <.0001      <.0001      <.0001
    2     <.0001                  0.1854      <.0001      0.0381      0.0101
    3     <.0001      0.1854                  0.0021      0.9918      0.8559
    4     <.0001      <.0001      0.0021                  0.0037      0.0142
    5     <.0001      0.0381      0.9918      0.0037                  0.9884
    6     <.0001      0.0101      0.8559      0.0142      0.9884
```

For comparison, here is the table of adjusted p-values for the Tukey method.

```
 i/j           1           2           3           4           5           6

    1                 <.0001      <.0001      <.0001      <.0001      <.0001
    2     <.0001                  0.0838      <.0001      0.0122      0.0026
    3     <.0001      0.0838                  0.0005      0.9808      0.7392
    4     <.0001      <.0001      0.0005                  0.0008      0.0039
    5     <.0001      0.0122      0.9808      0.0008                  0.9732
    6     <.0001      0.0026      0.7392      0.0039      0.9732
```

You can see that the Tukey p-values are almost all smaller than the Scheffé p-values, except when the values are near one. This is to be expected; the Tukey method is theoretically more powerful because the sample sizes are almost equal. Still, the two methods point to exactly the same conclusions for these particular data (and so does the Bonferroni method).

How would you *describe* these conclusions? This is the answer to the standard question "Which means are different from each other?" or just "What do you conclude?" If the question asks for "plain, non-statistical language," then you don't mention the multiple comparison method at all. Otherwise, you should add something like "These conclusions are based on a set of Bonferroni multiple comparisons using a joint 0.05 significance level."

But how much detail do you give, and what do you say? You can see that the Tables of adjusted p-values may be almost okay for a technical audience, but one can do a lot better. Here is an example. The format is based on one that SAS produces in connection with some multiple comparison methods you seldom want to do. Curiously, it is not available with `lsmeans`. I started by editing the list of means from `lsmeans` to put them in numerical order.

The table below shows mean length on the evening of day 10. Means that are not significantly different by a Scheffé test are connected by a common letter.

```
mcg      Mean Length on Day 10 (pm)

198      28.3250000
205      25.8500000  a
213      25.0000000  a  b
223      24.8000000     b
225      24.6000000     b
221      23.4000000
```

Here are the conclusions in plain language.

1. `mcg` 198 grows fastest.

2. `mcg` 221 grows slowest.

3. We cannot conclude that the growth rates of `mcg`s 205 and 213 are different.

4. `mcg` 205 grows faster than `mcg`s 221, 223 and 225.

5. `mcg` 213 grows faster than 221, but there is not enough evidence to conclude that it is different from 223 or 225.

6. There is little difference between the growth rates of `mcg`s 223 and 225.

This example illustrates something that can be a source of discomfort. The conclusions of multiple significance tests, even when they are multiple comparisons, need not be logically consistent with one another. Here, growth for mcg 205 is not different from 213, and 213 is not different from 223 — but 205 *is* different from 223. All I can say is that it would be worse if you were formally accepting the null hypothesis. Another weird thing is that it's mathematically possible for the overall F test to be significant, so you conclude that the population means are not all equal. But then *none* of the pairwise comparisons are significant, no matter what multiple comparison method you use. Ouch.

If you plan to use Scheffé's method to test contrasts other than (or in addition to) pairwise comparisons, it helps to have the adjusted critical value in front of you. Then you can just compare the F values from your `contrast` statements to the critical value. You could do it with a table of the F distribution and a calculator, but `proc iml` (which stands for "Interactive Matrix Language," and is very powerful) is more convenient, because the critical value appears on your output. Here is the code.

```
proc iml;
     title3 'Scheffe critical value for all possible contrasts';
     numdf = 5;   /* Numerator degrees of freedom for initial test */
```

```
dendf = 17;  /* Denominator degrees of freedom for initial test */
alpha = 0.05;
critval = finv(1-alpha,numdf,dendf);
scrit = critval * numdf;

print "Initial test has"  numdf " and " dendf "degrees of freedom."
      "----------------------------------------------------------"
      "Using significance level alpha = " alpha
      "------------------------------------------------"
      "Critical value for the initial test is " critval
      "------------------------------------------------"
      "Critical value for Scheffe tests is " scrit
      "------------------------------------------------";
```

And here is the output.

```
       Scheffe critical value for all possible contrasts

                     numdf           dendf

Initial test has         5  and         17 degrees of freedom.
  ----------------------------------------------------------
                                                alpha

          Using significance level alpha =       0.05
       ------------------------------------------------
                                                 critval

       Critical value for the initial test is  2.8099962
       ------------------------------------------------
                                                 scrit

        Critical value for Scheffe tests is  14.049981
       ------------------------------------------------
```

3.4.5 Scheffé tests for *collections* of contrasts

Scheffé tests actually protect a family of tests that include tests for infinitely many *collections* of contrasts, not just single contrasts. Suppose the initial F test is significant, and you have a follow-up null hypothesis saying that s non-redundant[13] contrasts all equal zero. In the `tubes` example, such a null hypothesis would be that the population means for all MCGs except 198 are equal – in other words, the test of whether the MCGs other

[13] Linearly independent.

than 198 have different growth rates. This involves $s = 4$ contrasts. We did it as a one-at-a-time test in `tubes09f.sas`; the contrast was named `AllBut198`.

To convert such a "planned" comparison to a Scheffé test, just use the adjusted critical value

$$f_{Sch} = f_{crit}\frac{p-1}{s}, \tag{3.1}$$

where f_{crit} is the usual critical value for the initial test. Then, considered as a Scheffé follow-up, the test is significant at the *joint* 0.05 level if the computed value of F for the collection of contrasts is greater than f_{Sch}.

For the example of `AllBut198`, $f_{crit} = 2.81, p = 6$ and $s = 4$. So

$$f_{Sch} = 2.81\frac{5}{4} = 3.51.$$

The test we got from `contrast` gave us $F = 21.44$, which is bigger than 3.51. So we conclude that those other growth rates are not all equal.

If you plan to test collections of contrasts with Scheffé tests, it is helpful to have a table of all the adjusted critical values you might need. Here is a `proc iml` that does the job. The details are not explained, but the code can easily be adapted to fit any example. All you need are the numerator degrees of freedom $(p-1)$ and denominator degrees of freedom $(n-p)$ from an ANOVA summary table.

```
proc iml;
title3 'Table of Scheffe critical values for COLLECTIONS of contrasts';
     numdf = 5;   /* Numerator degrees of freedom for initial test */
     dendf = 17;  /* Denominator degrees of freedom for initial test */
     alpha = 0.05;
     critval = finv(1-alpha,numdf,dendf);
     zero = {0 0}; S_table = repeat(zero,numdf,1);  /* Make empty matrix */
     /* Label the columns */
     namz = {"Number of Contrasts in followup test"
             "    Scheffe Critical Value"};
     mattrib S_table colname=namz;
     do i = 1 to numdf;
        s_table(|i,1|) = i;
        s_table(|i,2|) = numdf/i *  critval;
     end;
     reset noname; /* Makes output look nicer in this case */
     print "Initial test has"  numdf " and " dendf "degrees of freedom."
           "Using significance level alpha = " alpha;
     print s_table;
```

Here is the output.

```
          Table of Scheffe critical values for COLLECTIONS of contrasts

         Initial test has         5   and         17 degrees of freedom.
                   Using significance level alpha =      0.05


         Number of Contrasts in followup test     Scheffe Critical Value

                                            1                  14.049981
                                            2                  7.0249904
                                            3                   4.683327
                                            4                  3.5124952
                                            5                  2.8099962
```

When you do Scheffé tests for collections of contrasts, several comforting rules apply.

- If the initial test is not significant, it's a mathematical fact that no test for a collection of contrasts can be significant by a Scheffé test, so don't even bother.

- Suppose the Scheffé test for a collection is significant. Now consider the collection of all single contrasts that are equal to zero if all members of the collection equal zero[14]. The Scheffé test for at least one of those contrasts will be significant — if you can find it.

- Suppose the Scheffé test for a collection of s contrasts is *not* significant. If the truth of H_0 for the collection implies that a contrast is equal to zero, then the Scheffé test for that contrast cannot be significant either.

- The last point applies to smaller collections of contrasts, that is, to collections involving fewer than s contrasts.

3.4.6 Proper Follow-ups

We will describe a set of tests as *proper follow-ups* to to an initial test if

1. The null hypothesis of the initial test logically implies the null hypotheses of all the tests in the follow-up set.

2. All the tests are jointly protected against Type I error (false significance) at a known significance level, usually $\alpha = 0.05$.

The first property requires explanation. First, consider that the Tukey tests, which are limited to pairwise differences between means, automatically satisfy this, because if all

[14] Technically, the set of all vectors of weights that lie in the linear subspace spanned by the weights of the collection.

the population means are equal, then each pair is equal to each other. But it's possible to make mistakes with Bonferroni and Scheffé if you're not careful.

Here's why the first property is important. Suppose the null hypothesis of a follow-up test *does* follow logically from the null hypothesis of the initial test. Then, if the null hypothesis of the follow-up is false (there's really something going on), then the null hypothesis of the initial test must be incorrect too, and this is one way in which the initial null hypothesis is false. Thus if we correctly reject the follow-up null hypothesis, we have uncovered one of the ways in which the initial null hypothesis is false. In other words, we have (partly, perhaps) identified where the initial effect comes from.

On the other hand, if the null hypothesis of a potential follow-up test is *not* implied by the null hypothesis of the initial test, then the truth or untruth of the follow-up null hypothesis does not tell us *anything* about the null hypothesis of the initial test. They are in different domains. For example, suppose we conclude $2\mu_1$ is different from $3\mu_2$. Great, but if we want to know how the statement $\mu_1 = \mu_2 = \mu_3$ might be wrong, it's irrelevant.

If you stick to testing contrasts as a follow-up to a one-way ANOVA, you're fine. This is because if a set of population means are all equal, then any contrast of those means is equal to zero. That is, the null hypothesis of the initial test automatically implies the null hypotheses of any potential follow-up test, and everything is okay. Furthermore, if you try to specify a linear combination that is not a contrast with the `contrast` statement of `proc glm`, SAS will just say something like `NOTE: CONTRAST SOandSO is not estimable` in the log file. There is no other error message or warning; the test just does not appear in your list file.