

Lab 8: Inference in one-way ANOVA
STT 422: Summer, 2004
Vince Melfi

In this lab we'll learn about SAS procedures for comparing population means in the context of a one-way analysis of variance. An enormous number of techniques for comparing means are implemented in SAS; we certainly won't learn about all of these here, but information is available in the SAS help system.

To illustrate the methods, we'll use a data set from *Applied Linear Statistical Models* by Neter et. al. The data were collected to compare the effect of four different package designs on the sales of breakfast cereal. There are two variables of interest: **sales**, which measures the number of cases of breakfast cereal sold; and **design**, which indicates which of the four package designs was used.

Inference for individual means

The following program reads in and prints the data and computes confidence intervals for the four population means using several different methods. Explanation follows the program.

```
data cereal;
  infile 'u:\msu\course\stt\422\summer04\cereal.dat';
  input sales design;

proc print data = cereal;

proc means data = cereal
  mean std stderr clm alpha = 0.1 maxdec = 4;
class design;
var sales;

proc glm data = cereal;
  class design;
model sales = design;
means design/t clm alpha = 0.1;
  means design/bon clm alpha = 0.1;

run;
quit;
```

Explanation

1. **proc means** is a basic SAS procedure that computes descriptive statistics, confidence intervals, etc. The first three options **mean** **std** **stderr** tell SAS that we'd like to see the mean, standard deviation, and standard error. The next two options **clm** **alpha = 0.1** tell SAS that we'd like confidence limits, with $\alpha = 0.1$, which corresponds to a confidence level of $1 - \alpha = 0.90$. The option **maxdec = 4** tells SAS to print a maximum

of four decimal points. This just cleans up the output a bit. The confidence interval for μ_i is computed based only on the data from population i . For example, the confidence interval for the mean sales for design 1 is computed as

$$\bar{y}_{1.} \pm t^*(S/\sqrt{n_1}),$$

where S is computed from the five sales values for design 1, and t^* is computed from a t distribution with $n_1 - 1 = 4$ degrees of freedom. The statement `class design` tells SAS we'd like separate confidence intervals for each of the four designs, and `var sales` tells SAS for which variable we're interested in confidence intervals.

- Next we compute confidence intervals using `proc glm`. First we specify the model as usual. The `means` statements tell SAS we want information about the means. The statement `means design` tells SAS we want information about the means for each of the design types. After the forward slash we specify the method to use, in this example either `t` or `bon`, the fact that we want confidence limits `clm`, and the confidence level, again specified as 90% by stating `alpha = 0.1`.

The `t` method computes confidence intervals via

$$\bar{y}_i \pm t^*(MSE/sqrtn_i),$$

where t^* is the 95th percentile from a t distribution with $n - I = 19 - 4$ degrees of freedom. This method has no adjustment for multiple comparisons.

The `bon` method computes confidence intervals via a formula that looks similar:

$$\bar{y}_i \pm t^*(MSE/sqrtn_i).$$

Here again the t distribution with $n - I = 19 - 4$ degrees of freedom is used, but the percentile is adjusted using the Bonferroni method to take into account the multiple comparisons. In this case, since there are 4 means, the $\alpha = 0.1$ specification gets changed to $0.1/4 = 0.025$.

- Note that since the `proc glm` method uses all the data to estimate σ^2 via the MSE, which leads to a higher number of degrees of freedom, it will tend to yield shorter confidence intervals. But note also that this method is valid only if the population standard deviations are all the same, since this is the basis for pooling the data to compute the MSE.

Inference for pairwise differences of means

Next we look at inference for pairwise differences $\mu_i - \mu_k$. Here multiplicity adjustment becomes a more serious issue, since there are $I(I - 1)/2$ different comparisons to be made. (In our example this means there are 6 comparisons. Run the following program.

```
proc glm data = cereal;
  class design;
  model sales = design;
```

```
means design/t tukey bon scheffe;
```

```
run;  
quit;
```

In this program we don't run the equivalent of the **proc means** statement in the previous program. The syntax for **proc glm** is similar to the earlier program, except the `clm` option is omitted, and two more multiple comparison adjustment methods are included, the Tukey and the Scheffe methods.

Look at the output to see the effects of the various comparison methods.¹ Compare, for example, the confidence interval for $\mu_4 - \mu_1$. Remember that the `t` method does not adjust for multiplicity, so its confidence intervals are suspect. The Bonferroni and Scheffe methods will tend to give too wide (conservative) intervals in comparing all means, since they weren't designed specifically for this purpose.

Contrasts

The `contrast` statement and the `estimate` statement perform hypothesis tests and estimation for contrasts. We'll focus on using `estimate` to compute estimates and confidence intervals for contrasts. Consider the following program.

```
proc glm data = cereal;  
  class design;  
  model sales = design / clparm;  
  estimate '1&2 vs 3&4' design .5 .5 -.5 -.5;
```

```
proc means data = cereal;  
  var sales;  
  class design;
```

```
run;
```

There are a few new things in this program. First, the option `clparm` is given as part of the model statement. This indicates that confidence intervals are desired for the results of all `estimate` statements. Second, the `estimate` statement indicates the particular contrast we'd like to estimate. In this example, the contrast has values $c_1 = 0.5, c_2 = 0.5, c_3 = -0.5, c_4 = -0.5$. **proc means** was run at the end to facilitate checking the contrast estimate by hand. A label enclosed in " is required as part of the `estimate` statement.

Note that the confidence interval produced as part of the `estimate` statement does not make use of any multiplicity adjustment. This seems odd. I'm not sure how to convince SAS that a multiplicity adjustment is desired for contrasts.

¹For some reason SAS prints $I(I-1)$ comparisons, half of which are redundant. For example, the interval for $\mu_4 - \mu_1$ gives the same information as the interval for $\mu_1 - \mu_4$.