

An Interactive Approach to Multiple Comparison Procedures

Peter P. Davis, U.S. Census Bureau, Washington, DC¹

ABSTRACT

The area of multiple comparison procedures is complex and large. Many SAS procedures such as PROC GLM and PROC MULTTEST, just to name a few, are available to handle multiple inferences depending upon the data and the inference objectives. However, often in research, a quick assessment to detect differences among groups where the standard errors are known is necessary. Therefore, this paper presents a simple, interactive, multiple comparison test using %WINDOW and %DISPLAY. The user specifies N , the number of groups or treatments and α , the significance level. Then, the user is prompted N times for each group estimate and standard error. Finally, test results are available for all pairwise comparisons.

INTRODUCTION

Multiple comparisons and multiple tests frequently arise in all areas of data analysis. Often when analyzing data within a single study, there is a need to compare more than two groups. The question asked is, "Does a difference among the groups exist?" A simple one-way ANOVA test for the equality of group means answers this question. In SAS, this is accomplished using PROC GLM. However, a more informative question would be, "What differences between the groups actually exist?" More specifically, when compared two at a time, which groups are significantly different?

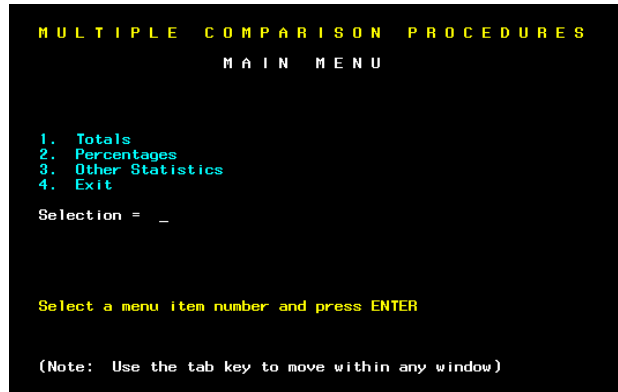
Suppose a team of researchers has completed their work on a sample survey. Assume survey estimates have been produced and their standard errors have been calculated using some variance estimation technique. As a final stage of the analysis, a quick assessment is needed to determine what differences, if any, exist among the groups within the study.

This paper demonstrates a menu-driven multiple comparison program designed to investigate differences among groups within a study. Two important features of the program include: (1) we use the Bonferroni method for multiple comparisons and (2) we assume independence ignoring any possible correlation among estimates. At various stages of the program, the user is prompted to enter the information needed to conduct the multiple comparison procedure. This program only addresses two-tailed hypothesis tests on all pairwise comparisons between statistics where the standard errors are known. Finally, the program can be used to quickly assess differences among groups paired two at a time.

To facilitate an interactive environment, we make use of the %WINDOW and %DISPLAY statements that are controlled by the macro processor. The %WINDOW statement defines a macro window. The %DISPLAY statement displays the macro window. Once a window is developed and customized, these windows can be used to display text or accept input as is done in this application which is designed to conduct a multiple comparison test.

CUSTOMIZING WINDOWS

The application begins with a main menu. See Figure 1. In the main menu window, several multiple comparison procedures are listed: totals, percentages, and other statistics. The user is prompted to choose a multiple comparison procedure suitable to the statistic of interest or they may choose to exit the program.



```
MULTIPLE COMPARISON PROCEDURES
MAIN MENU

1. Totals
2. Percentages
3. Other Statistics
4. Exit

Selection = _

Select a menu item number and press ENTER

(Note: Use the tab key to move within any window)
```

Figure 1: Main Menu Window

As seen in the code below, the main menu window is defined and customized by the %WINDOW statement. The %WINDOW statement gives a name to the current window, mainmenu, lists the attributes of the window, and describes the purpose of the window: to display text, to accept user input, or both.

```
/* Define Main Menu */
%window mainmenu color=black
#5 @15 "MULTIPLE COMPARISON
PROCEDURES" color=yellow
#7 @35 "MAIN MENU" color=white
#12 @15 "1. Totals" color=cyan
#13 @15 "2. Percentages" color=cyan
#14 @15 "3. Other Statistics" color=cyan
#15 @15 "4. Exit" color=cyan
#17 @15 "Selection = " color=white +1 choice 1
color=white attr=underline
#23 @15 "Select a menu item number and press
ENTER" color=yellow
#27 @15 "(Note: Use the tab key to move within
any window)" color=white;
```

Each line of the %WINDOW statement represents a characteristic of the window. The window name, mainmenu, immediately follows the %WINDOW statement. In addition, using only one of many window options, we have specified the window's background color to be black. The COLOR = option achieves this desired result.

¹ The author is a mathematical statistician in the Decennial Statistical Studies Division, U.S. Census Bureau. This paper reports the results of research and analysis undertaken by Census Bureau staff. It has undergone a Census Bureau review more limited in scope than that given to official Census Bureau publications. This report is released to inform interested parties of ongoing research and to encourage discussion of work in progress.

Text appearing in quotes in the main menu window is displayed on the screen. For example, the first line contains "MULTIPLE COMPARISON PROCEDURES." This becomes the first line in the title of the main menu. Where this text is displayed on the screen is defined by two location indicators, the pound sign (#) and the at sign (@). The pound sign indicates the line number and the at sign indicates the column. The first line of the title, "MULTIPLE COMPARISON PROCEDURES," begins on line 5 (#5) and at column 15 (@15).

Several display attributes for any text field are available. Similar to the window background, the color of text displayed on the screen can be designated. For example, the color of a field is specified by the COLOR = option. The field "1. Totals" is colored cyan. Another display feature is the ATTR = option. In this window, we have specified the underline attribute, ATTR = underline. This feature underlines a field.

In addition to displaying text, the %WINDOW statement has the ability to read user input. The above %WINDOW statement contains the following line:

```
#17 @15 "Selection = " color=white +1 choice 1
color=white attr=underline
```

This line within the %WINDOW statement creates a macro variable, CHOICE. At line 17 and starting at column 15, the text "Selection = " is displayed. The macro variable CHOICE is positioned one space to the right (+1). The number 1 following the variable restricts the length of the macro variable CHOICE to one character. Also, notice CHOICE is blank. This is because no previous value has been given to this variable. The user uses the tab key to move the cursor to this location to make a selection. Then, the user types the number of the desired multiple comparison task in the selection field and presses enter. After pressing enter, the value typed is stored in the macro variable, CHOICE. Once a macro variable has been assigned a value, that value is displayed in the window. If the window is displayed again without resetting the variable back to null or blank, the variable's previously defined value will appear in the window. Restarting the SAS session will also clear any previously defined macro variables.

After customizing the macro window mainmenu, it is displayed using the %DISPLAY statement:

```
/* Display Main Menu */
%display mainmenu;
```

Once a macro window is defined, it can be displayed at any point and as many times as needed simply by referring to its window name. Once created, a macro window exists until the SAS session ends. Therefore, for windows whose definitions do not change, it is more efficient to define these unchanging windows outside a macro or in a macro that executes only once.

The windows in this application follow a top-down structure or a parent-child relationship. The window entitled mainmenu is the parent window of the entire multiple comparison procedure application. Any subsequent window of mainmenu is deemed a child window. Keep in mind, as you develop applications which are more complex, a window may be a parent of one window and simultaneously be a child of another window.

CHILD WINDOWS and USER INPUT

The window entitled mainmenu creates three child windows, one for each of the three multiple comparison procedures listed in the

main menu window. For the purposes of this paper, let's focus on the last one, multiple comparison tests for "Other Statistics." The window created by choosing this test is seen in Figure 2.

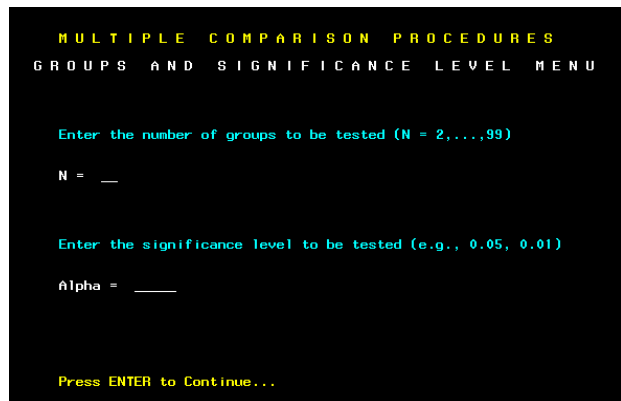


Figure 2: Other Statistics Window

Arriving at this window is based on the choice you make in the mainmenu window. To arrive at this window, the user typed "3" at the main menu and pressed enter. Choosing "1" or "2" would have resulted in a different child window leading to another multiple comparison procedure.

The %WINDOW statement for the "Other Statistics" window and its %DISPLAY statement are shown below. This macro window is given the name ntests. Similar to mainmenu, this window, ntests, prompts the user for values of two macro variables: N and ALPHA. N is the number of groups or treatments in the study and ALPHA is the level of significance. Using the tab key again, you can enter values for N and ALPHA. Notice in the code the number after the macro variable N and ALPHA. Again, these numbers denote the length of the variable being created. The length of N is two and ALPHA is of length five. Therefore, this multiple comparison procedure can run up to 99 groups. ALPHA can be set to any level of significance needed: 0.01, 0.025, 0.05, 0.01, etc. Setting the field length of N is used as a control to limit the number of groups because too many would be unrealistic. Limiting the size of ALPHA to five characters allows for varying levels of significance.

```
/* Define Groupings/Alpha Menu */
%window ntests color=black
#5 @15 "MULTIPLE COMPARISON
PROCEDURES" color=yellow
#7 @12 "GROUPS AND SIGNIFICAN
CE LEVEL MENU" color=white
#12 @15 "Enter the number of groups to be tested
(N = 2,...,99)" color=cyan
#15 @15 "N = " color=white +1 n 2 color=white
attr=underline required=yes
#20 @15 "Enter the significance level to be tested
(e.g., 0.05, 0.01)" color=cyan
#23 @15 "Alpha = " color=white +1 alpha 5
color=white attr=underline required=yes
#30 @15 "Press ENTER to Continue..."
color=yellow;
```

```
/* Display Groupings/Alpha Menu */
%display ntests;
```

A new feature added in this %WINDOW statement is the REQUIRED = option. The number of groups to be tested, N, and the significance level, ALPHA, are required fields. This means a value must be entered for these macro variables. The application will not continue without values for these variables. Since N and ALPHA are critical values necessary in conducting multiple comparison tests, we can designate these as required fields by specifying REQUIRED = YES. Once a value for these variables has been entered, the application continues after pressing ENTER.

DATA ENTRY

At some point, it had to come along--data entry. At this point in the multiple comparison procedure application, three macro variables exist. The first one provided us with a choice from the main menu leading us down the path to conduct a multiple comparison test for "other statistics" when the standard error is known. The other two variables are N, the number of groups in the study, and ALPHA, the level of significance for the test.

To conduct this appropriate multiple comparison test, we need to know the value of the estimate and its corresponding standard error for all N groups. Hence, we need a data entry facility to capture and store each estimate and standard error in order to perform a multiple comparison test.

The code below serves two purposes: (1) it displays a generic macro window designed to obtain the value of an estimate and its standard error and (2) it repeats or redisplay the window N times to capture all estimates and all standard errors.

```
%macro repeatn;
```

```
/* Define Estimates and Standard Error Menu.
This Menu repeats N times, one for each Group.*/
%do i = 1 %to &n;
%global est&i se&i;
%window estse color=black
#5 @15 "MULTIPLE COMPARISON
PROCEDURES" color=yellow
#7 @22 "ESTIMATES MENU - GROUP
&i" color=white
#12 @15 "Enter the following for Group &i"
color=cyan
#15 @15 "Estimate for Group &i = " color=white +2
est&i 5 color=white attr=underline
#16 @15 "Standard Error for Group &i = "
color=white +2 se&i 5 color=white attr=underline
#23 @15 "Press ENTER to Continue..."
color=yellow;
%display estse;
%end;

%mend repeatn;
%repeatn
```

In the center of this code is the %WINDOW statement which creates a data entry menu. This macro window, called estse, is defined within a macro. The macro is repeated N times, once for

each group in the study. The unique item of interest in this window is that certain items displayed in the window on the screen change with each group. For example, in this macro window, the text "ESTIMATES MENU - GROUP &i" will be labeled "ESTIMATES MENU - GROUP 1" for the first group in the study, "ESTIMATES MENU - GROUP 2" for the second group in the study, and so on. Likewise, the macro variables EST1 through EST<N> and SE1 through SE<N> are created to store the estimates and standard errors for each group. Figure 3 shows the estse window for Group 2.

Figure 3: Estimates Window

From the above code, a basic data entry process has been created using a macro window and a few other macro statements to repeat the macro window as many times as necessary. All the necessary data to perform a multiple comparison test has been obtained and stored as macro variables.

UTILIZING USER INPUT

Once all the pertinent data has been entered, the appropriate multiple comparison test can be performed. Following the example from above, we choose option three at the main menu. This is a multiple comparison test for other statistics where the standard error is known. The data step in the code below executes this desired multiple comparison test.

```
%macro assign;
data test (drop=est1 - est&n se1 - se&n)
siglev (keep=ncomp adjalpha grpa grpb
teststat pval sig);
array est[&n.] est1 - est&n;
array se[&n.] se1 - se&n;

%do i = 1 %to &n;
est[&i.]=&est&i;
se[&i.]=&se&i;
%end;

ncomp=gamma(&n+1)/(gamma(&n-
1)*gamma(3));

adjalpha = &alpha/ncomp;
```

```

%do i = 1 %to &n;
  %do j = %eval(&i+1) %to &n;
    stat&i&j = (est&i -
est&j)/sqrt((se&i**2) + (se&j**2));
    pval&i&j = 2 * (1 -
probnorm(abs(stat&i&j)));

    if pval&i&j < adjalph then do;
      sig&i&j = 'Y';
    end;
    else do;
      sig&i&j = 'N';
    end;
    output test;

    grpa = &i;
    grpb = &j;
    teststat = stat&i&j;
    pval = pval&i&j;
    sig = sig&i&j;

    output siglev;
  %end;
%end;
run;
%mend assign;
%assign

```

The primary goal of this data step is to take the values entered as macro variables in the previous windows and assign them to arrays so that the data is more manageable. In this data step, the estimates are assigned into one array and the standard errors are assigned into another array.

Using the number of groups in the study and knowing that pairwise comparisons will be made, the number of comparisons is calculated taking N items, two at a time, without replacement, or

as seen in some statistics books, the symbol $\binom{n}{2}$, read “n

choose 2.” When using the Bonferroni technique for multiple comparisons, an adjusted alpha is calculated to determine true differences, simultaneously, for all pairwise comparisons. This adjusted alpha is the significance level divided by the number of comparisons. Thus, the probability of making a Type I error, or falsely rejecting the null hypothesis, for each test is the adjusted alpha which results in an overall probability of a Type I error equal to ALPHA, the level of significance.

A macro performs all “n choose 2” tests. For each comparison, an observed test statistic is calculated along with its p-value. A comparison is flagged as “significant” if the p-value for the observed test statistic is less than the adjusted p-value.

RESULTS

Results of the multiple comparison procedure are printed using a simple PROC PRINT. The PROC PRINT provides results in a list format. All pairwise comparisons are displayed with their test statistic, p-value, and whether or not a difference is deemed significant.

CONCLUSIONS

This paper demonstrates a simple menu-driven approach to multiple comparison procedures. Only the basics of creating menus, capturing data, converting data for use, and displaying test results is addressed. Although this application is not the most sophisticated, it does achieve a quick and easy result for detecting any differences between groups within a study.

ENHANCEMENTS

Other items that might be of interest to enhance this menu include:

1. A “Return to Main Menu” button which would be included on any menu other than the main menu. Pressing this button would send the user back to the beginning and set all previously defined variables to blank or null.
2. A check to verify that a valid number has been entered by the user. For example, at the main menu, entering a number greater than 4 would produce an “error” window.
3. A “Clear the Screen” button which would clear the screen and set all variables for only that screen to blank.
4. Changing the color of each macro window so that when a user arrived at a different menu, it would be clearly visible.

REFERENCES

SAS[®] *Macro Facility Tips and Techniques, Version 6, First Edition*, Cary, NC: SAS Institute Inc., 1994. 319 pp.

SAS Institute Inc., *SAS Macro Language: Reference, Version 8*, Cary, NC: SAS Institute Inc., 1999. 310 pages.

Westfall, Peter H., Tobias, Randall D., Rom, Dror, Wolfinger, Russell D., and Hochberg, Yosef, *Multiple Comparisons and Multiple Tests Using the SAS[®] System*, Cary, NC: SAS Institute Inc., 1999. 416 pp.

SAS[®] and all other SAS Institute Inc. Product or service names are registered trademarks or trademarks of SAS institute Inc. in the USA and other countries. © indicates USA registration.

CONTACT INFORMATION

Contact the author at:

Peter Davis
U.S. Census Bureau
4700 Silver Hill Road, Rm 2505-2
Washington, DC 20233-7600
Tel: 301.763.4291
Fax: 301.457.4580
Email: peter.p.davis@census.gov
Web: www.census.gov