

Huixing Tang & Matthew Grover  
The Psychological Corporation  
Paper presented at the Third Annual South-Central  
Regional SAS Users Group Conference , Sept. 1993

## **Automating the Plotting of Significant Effects in Analysis of Variance**

### **I. INTRODUCTION**

When running analysis of variance involving one or more independent variables, one may often wish to plot the data to help interpret the statistical results. A graphic display is particularly important for interpreting significant interaction effects. For example, researchers often rely on the plot of the means at different levels of the independent variables to determine whether an interaction is ordinal or disordinal. Such information is essential for interpreting interaction effects as well as main effects.

The current SAS procedures (e.g., PROC GLM\ANOVA) for analysis of variance do not provide automatic ways of plotting significant effects. One common method to deal with this limitation is to generate the statistics, inspect the results, and then plot the results in a separate step. While this may not present any problem for small scale data analysis, it can be quite burdensome and inefficient for large scale analysis, especially when it involves a large number of data sets or variables. Imagine that three-way analysis of variance is run on a set of data. When the three-way interaction is significant, the researcher may want to plot the two-way interactions at each level of a third variable to see the extent to which the two-way interactions are different across levels. Each of the three independent variables could be used as a 'by variable', and for each two-way interaction, there are two ways of plotting the data depending on which variable is used as the 'horizontal variable'. Thus, there are  $3 \times 2$  (6) possible ways of plotting the three-way interaction effect alone. If analysis needs to be repeated for a large number of data sets and several different dependent variables, it can be, indeed, very cumbersome if the plotting has to be done separately with different code written for plotting every significant effect.

In situations similar to the one described above, one may wish to have an automated procedure by which graphic plots are processed in the same job as the statistical analysis. Even more desirable is an automated procedure in which graphic processing interacts with statistical processing such that only the data that meet certain statistical criteria are processed for graphic display. The efficiency of the procedure can be further enhanced if generic code is available. This will make it very easy for the researcher to specify which of the possible ways the data should be plotted.

The purpose of this paper is to present a procedure to automate the plotting of significant effects following an analysis of variance. The main feature of the procedure is that statistical computing and graphic processing are accomplished simultaneously and that the plotting of effects is data-driven according to user-specified statistical criterion. Another important aspect of the procedure is that generic code is written where possible to enhance flexibility in user-specifications of the criterion value, variable names, and other plotting options. The second section of this paper outlines the procedure used, followed by the use of the procedure with SAS base software and SAS/GRAPH. Finally the benefits of the procedure are discussed.

## II. OUTLINE OF THE PROCEDURE

The procedure for automated plotting consists of the following steps:

1. Create macro variables for all possible effects, the dependent variable. Specify the critical value below which plot code is produced.
2. Compute the means and standard errors at different levels of the independent variables and write them to an output data set.
3. Run ANOVA or GLM procedure and save the significance values in an output data set.
4. Use the significance value data obtained in step 3 as input data to generate SAS code for plotting. The user-specified criterion values declared in step 1 will determine which effects to plot and which data sets to use. The resulting SAS code is written out to an external file.
5. Execute the SAS code generated in step 4 using the *%include* statement. The results are the graphic plots generated according to pre-specified statistical criteria (see Figure 1 for a graphic presentation of the procedure).

## III. PLOTTING WITH SAS/BASE SOFTWARE

This section describes how the procedure outlined in the preceding section can be used with SAS/BASE software. For illustration, the procedure is applied to a factorial analysis. The purpose of the study is to analyze the effect of gender, ethnicity, and socio-economic status (SES) on student performance on a set of performance-based mathematics items. The dependent variable is the residual item scores (RES), scores that have been corrected for ability. The independent variables are gender, ethnicity, and SES. The data include eleven grade levels with a different set of items being used at each level. The purpose of the application is to perform a three-way analysis of variance on RES by level and item, and generate main-effect and/or

interaction effect plots when the effect is significant. A step by step illustration of the code used is given below.

The first step involves assigning libraries, creating macro variables to denote all the possible effects to be tested, and specifying the probability value used as the critical value. To prevent concatenated variable names which will also be used as data set names from exceeding 8 characters, only the first character of the variable name is used (E, G, and S stand for ethnic, gender, and SES, respectively). Macro variables *&VAR1--&VAR7* refer to all the possible combinations of the three independent variables. *&EFFECT1--&EFFECT7* are created for concatenated variable names and also data set names.

***/\* Step 1: Create macro variables for all possible effects, the dependent variable and the probability value \*/***

*options nobyline;*

*filename dd 'sascon.libexec';  
libname t 'sascon.libsas';*

*%let critical=.05;                               \* set significance value;  
%let dv=res;                                    \* dependent variable;*

*%let var1=e;                                   \* independent variables;  
%let var2=g;  
%let var3=s;  
%let var4=&var1 &var2;  
%let var5=&var1 &var3;  
%let var6=&var2 &var3;  
%let var7=&var1 &var2 &var3;*

*%let effect1=&var1;  
%let effect2=&var2;  
%let effect3=&var3;  
%let effect4=&var1.\_&var2;  
%let effect5=&var1.\_&var3;  
%let effect6=&var2.\_&var3;  
%let effect7=&var1.\_&var2.\_&var3;*

The second step involves the creation of plotting data for all possible effects. The means, standard errors, and the upper and lower error bounds are computed for each of the seven effects and are saved in seven data sets referenced by *&effect1--&effect7*.

***/\* Step 2: Compute means, std errors, and error bounds by effect \*/***

```
%macro means;
%do i= 1 %to 7;      * 7 = the number of possible effects;

proc sort data=t.rawdata;
  by level item &&var&i;

proc means noprint data=t.rawdata;
  var &dv;
  by level item &&var&i;
  output out=&&eff&i(drop=_type_) mean=mean stderr=se;

  data &&eff&i;
    set &&eff&i;
      u=mean + (2 * se);      * compute upper bound;
      l=mean - (2 * se);      * compute lower bound;
  %end;
%mend;
%means;
```

The third step involves running PROC GLM and using the OUTSTAT = option to save the probability values in the data set *pvalue*.

***/\* Step 3: Run PROC GLM, save the prob values in a data set \*/***

```
proc glm data=t.rawdata noprint outstat=pvalue;
  class &var1 &var2 &var3;
  model res=&var1 &var2 &var3 &var1*&var2 &var1*&var3
    &var2*&var3 &var1*&var2*&var3;
  by level item;

proc transpose data=pvalue out=pvalue(drop=_name_);
  var prob;      * probability values;
  by level item;
  id _source_;
  where _type_='SS3';
```

The *pvalue* data set contains the probability values associated with each possible effect for all items at every level. The contents of the *pvalue* data set are shown below.

| LEVEL | ITEM | E    | G    | S    | E_G  | E_S  | G_S  | E_G_S |
|-------|------|------|------|------|------|------|------|-------|
| 1     | 1    | .059 | .962 | .219 | .300 | .822 | .505 | .260  |
| 1     | 2    | .058 | .894 | .000 | .177 | .016 | .752 | .367  |
| 1     | 3    | .021 | .546 | .002 | .905 | .128 | .627 | .135  |
| 1     | 4    | .320 | .027 | .266 | .144 | .153 | .181 | .400  |
| 1     | 5    | .016 | .367 | .325 | .799 | .323 | .379 | .710  |
| 1     | 6    | .046 | .759 | .040 | .430 | .572 | .312 | .658  |
| 1     | 7    | .038 | .588 | .052 | .487 | .134 | .971 | .848  |
| 1     | 8    | .000 | .181 | .001 | .044 | .013 | .596 | .901  |
| 1     | 9    | .073 | .776 | .101 | .196 | .102 | .693 | .453  |
| 1     | 10   | .201 | .584 | .358 | .506 | .950 | .090 | .459  |
| 2     | 1    | .912 | .001 | .073 | .001 | .051 | .017 | .000  |
| 2     | 2    | .167 | .133 | .594 | .175 | .963 | .252 | .114  |
| 2     | 3    | .338 | .670 | .074 | .445 | .907 | .838 | .463  |
| 2     | 4    | .000 | .036 | .005 | .009 | .000 | .090 | .110  |
| 2     | 5    | .820 | .807 | .052 | .619 | .027 | .475 | .681  |
| 2     | 6    | .058 | .491 | .184 | .986 | .389 | .413 | .740  |
| 2     | 7    | .998 | .362 | .479 | .667 | .456 | .898 | .872  |
| 2     | 8    | .758 | .980 | .418 | .660 | .851 | .761 | .871  |
| 2     | 9    | .617 | .862 | .154 | .409 | .570 | .732 | .831  |
| 2     | 10   | .842 | .805 | .170 | .617 | .660 | .885 | .624  |

The fourth step uses the *pvalue* data set as the input data to generate SAS code for plotting the data:

1. The three-way interaction effect is examined first. If it is significant, the two-way interactions at each level of a third variable is plotted. This is accomplished by the macro *THREeway*. The variable used as the 'by variable', and the variable used as the 'horizontal variable' can be specified or changed easily in the macro call statement.
2. If the three-way interaction is not significant, each of the two-way interactions are examined. If any of them are significant, plotting of the two-way interaction will be obtained by collapsing over the levels of the third variable. This is accomplished by the macro *TWOWAY*. Since there are three two-way interaction effects, the macro should be invoked three times.
3. If the main effect is significant and none of the interaction effects involving the main effect variable are significant, the means and error bounds for each level of the main effect are plotted.

All the code is written to an external file *sascon.libexec(plotcode)* to be executed in the next step.

```
/* specifying the external file to contain the plotting code */
```

```
data _null_;  
set pvalue;  
file dd(plotcode);
```

```
/* Step 4a: macro for generating code to plot three-way interactions */
```

```
%macro threeway (prob,vara,varb,byvar);
```

```
if &prob <=&critical then put
```

```
"proc sort data=&prob;"/  
"by &byvar;          "/
```

```
" proc plot data=&prob ;"/  
" plot mean* &vara = '*' $ &varb; "/  
" where item=" item "and level=" level " ;"/  
" by &byvar;          "/  
"title1'Simple &vara.&varb @ #byval(&byvar) ' ; "/  
"title2' Level="level", Item "item", P="&prob" ' ;  "/ ;
```

```
%mend threeway;
```

```
%threeway(&eff7,&var1,&var2,&var3);
```

Note that with the above macro call statement, &var3 is used as the 'by variable'. The two-way interaction of &var1 and &var2 will be plotted at each level of &var3. &var1 is used as the 'horizontal variable'. If a different variable is to be used as the 'by variable' and/or a different way of plotting the two-way interaction is desired, changes can be easily made by switching positions of &var1, &var2, and &var3 in the macro call statement.

```
/* Step 4b: macro for generating code to plot two-way interactions */
```

```
%macro twoway(prob,vara,varb);
```

```
if &eff7 >=&critical and &prob < &critical then put
```

```
"proc plot data=&prob ;"/  
" plot mean* &vara = '*' $ &varb; "/  
" where item=" item "and level=" level " ;"/  
"title1'Twoway Interaction of &vara and &varb ' ; "/  
"title2' Level="level", Item "item", P="&prob" ' ;  "/ ;  
%mend twoway;
```

```
%twoway(&eff4,&var1,&var2);
```

```

%twoway(&eff5,&var1,&var3);
%twoway(&eff6,&var2,&var3);

/* Step 4c: macro to generate code to plot the main effects */

%macro oneway(prob1,prob2,prob3,vara);

if &eff7 >=&critical then if (&prob1>= &critical and &prob2>= &critical)
  then if &prob3 <&critical then put

      " proc plot data=&prob3;"/
      " where item=" item ";"/
      " plot mean*&vara = '*' "/
      "      l*&vara = '-' "/
      "      u*&vara = '-'/ overlay ;"/
      " where item=" item "and level=" level " ;"/
      "title1' Main Effects for &vara ' ; "/
      "title2' Level="level", Item "item", P="&prob3" ' ;  "/ ;

%mend oneway;

%oneway(&eff4,&eff5,&eff1,&var1);
%oneway(&eff4,&eff6,&eff2,&var2);
%oneway(&eff5,&eff6,&eff3,&var3);

/* Step 5: execute the codes written to sascon.libexec(plotcode) */

%include dd(plotcode);

```

See Appendix 1 for an example of code generated in this step.

## VI. PLOTTING WITH SAS/GRAPH

Steps 1-3 for plotting with SAS/GRAPH are identical to steps 1-3 for plotting with SAS/BASE software. Therefore , the following code will start with step 4.

```
/* Specifying an external file to contain the plotting code */
```

```
data _null_;  
  set pvalue;  
  file dd(gplot);
```

```
/* Step 4: macro for generating code to plot three-way interactions */
```

```
%macro threeway (prob, vara, varb, byvar);
```

```
  if &prob <=&critical then put
```

```
    "symbol1 v=square h=5 l=1 w=2 i=join c=black; "/
```

```
    "symbol2 v=diamond h=6 l=34 w=2 i=join c=black; " /
```

```
    "proc sort data=&prob;"/
```

```
    "by &byvar;          "/
```

```
    " proc gplot data=&prob ;"/
```

```
    " plot mean *&vara=&varb "/
```

```
    " /legend=legend1 vaxis=axis1 vminor=1 haxis=axis2;"/
```

```
    " by &byvar;          "/
```

```
    " label &byvar='00'x; " /
```

```
    " where level="level" and item=" item ";"/
```

```
    "title1 h=3.5 'Simple &vara.&varb @ #byval(&byvar) ' ; " /
```

```
    "title2 h=3.5 ' Level="level", Item "item", P="&prob" ' ; " / ;
```

```
  %mend threeway;
```

```
%threeway(&eff7,&var1,&var2,&var3);
```

```
%threeway(&eff7,&var1,&var3,&var2);
```

```
%threeway(&eff7,&var2,&var3,&var1);
```

```
/* Step 4: macro for generating code to plot two-way interactions */
```

```
%macro twoway(prob, vara, varb);
```

```
if &eff7 >=&critical and &prob < &critical then put
```

```

"symbol1 v=square h=5 l=1 w=2 i=join c=black; "/
"symbol2 v=diamond h=6 l=34 w=2 i=join c=black; " /

"proc gplot data=&prob; "/
"plot mean*&vara=&varb"/
" /legend=legend1 vaxis=axis1 vminor=1 haxis=axis2; "/
" where level="level" and item=" item "; "/
"title1 h=3.5 'Twoway Interaction of &vara and &varb ' ; " /
"title2 h=3.5 ' Level="level", Item "item", P="&prob" ' ; " / ;

%mend twoway;

%twoway(&eff4,&var1,&var2);
%twoway(&eff5,&var1,&var3);
%twoway(&eff6,&var2,&var3);

/* Step 4: macro to generate code to plot the main effects */

%macro oneway(prob1,prob2,prob3,vara);

if &eff7 >=&critical then if (&prob1 >=&critical and &prob2 >=&critical) then
    if &prob3 <&critical then put

"symbol1 f=marker v=t h=4 w=2 i=none c=black; " /
"symbol2 f=music v=j h=3 w=2 i=none c=black; " /
"symbol3 f=music v=j h=3 w=2 i=none c=black; " /

" proc gplot data=&prob3; "/
" plot mean*&vara " /
" l*&vara " /
" u*&vara " /
" / vaxis=axis1 vminor=1 haxis=axis2 overlay; "/
" where level="level" and item=" item " ; " /
"title1 'Main Effects for &vara ' ; " /
"title2 ' Level="level", Item "item", P="&prob3" ' ; " /
"run; symbol1; symbol2; symbol3; " / ;

%mend oneway;

%oneway(&eff4,&eff5,&eff1,&var1);
%oneway(&eff4,&eff6,&eff2,&var2);
%oneway(&eff5,&eff6,&eff3,&var3);

axis1 label=(h=3 )
value=(font=simplex h=3.0)
offset=(2);

```

```

axis2 label=(h=4 ' ')
      value=(h=3)
      offset=(3)
      length=45
      minor=none;

legend1 label=(h=4 ' ')
       value=(h=3);
run;

```

*/\* Step 5: execute the code written to sascon.libexec(gplot) \*/*

```
%include dd (gplot); run;
```

See Appendix 2 for an excerpt of code generated in this step

## V. CONCLUSION

Multiple benefits can be realized from the procedure explored in this study. First, code to plot group means is written out to file only when the effects are significant, which is determined by a probability value preset by the user. The plots of significant effects are then produced with a *%include* statement which brings in the file that contains the plotting code. This way valuable time is not spent generating plots of effects that are not significant. The analysis can be repeated numerous times for different significance levels just by changing the value of the macro variable *&critical* which declares the critical probability value. Further, the generic nature of the code allows it to be applied to any set of data. By just changing the lines of code which declare the independent variables, the dependent variable, and the probability value new data can be accommodated. Thus, once the code has been set up it can be easily used and modified by even a novice SAS user. Total understanding of the code is not required to run analysis.

## APPENDIX 1

An excerpt of some of the codes that were written to *sascon.libexec(plotcode)*

```
proc plot data=e;
  plot mean*e = '*'
        l*e = '-'
        u*e = '-'/ overlay ;
  where level=1 and item=3;
  title1 'Main Effects for E ';
  title2 'Level=1, Item 3, P=0.212';

proc plot data=e_g ;
  plot mean* e = '*' $ g;
  where item=8 and level=1 ;
  title1 'Twoway Interaction of E and G ';
  title2 'Level=1, Item 8, P=0.043';

proc sort data=e_g_s;
  by g;

proc plot data=e_g_s ;
  plot mean* e = '*' $ s;
  where item=1 and level=2 ;
  by g;
  title1 'Simple ES @ G ' ;
  title2 ' Level=2, Item 1, P=0.000 ';
```

## APPENDIX 2

An excerpt of some of the code that were written out to *sascon.libexec(gplot)* .

```
symbol1 f=marker v=t h=4 w=2 i=none c=black;
symbol2 f=music v=j h=3 w=2 i=none c=black;
symbol3 f=music v=j h=3 w=2 i=none c=black;

proc gplot data=e;
  plot mean*e
    l*e
    u*e
  / vaxis=axis1 vminor=1 haxis=axis2 ) overlay;
  where level=1 and item=3 ;
  title1 h=3.5 'Main Effects for E ';
  title2 h=3.5 'Level=1, Item 3, P=0.212';

run; symbol1; symbol2; symbol3;

symbol1 v=square h=5 l=1 w=2 i=join c=black;
symbol2 v=diamond h=6 l=34 w=2 i=join c=black;

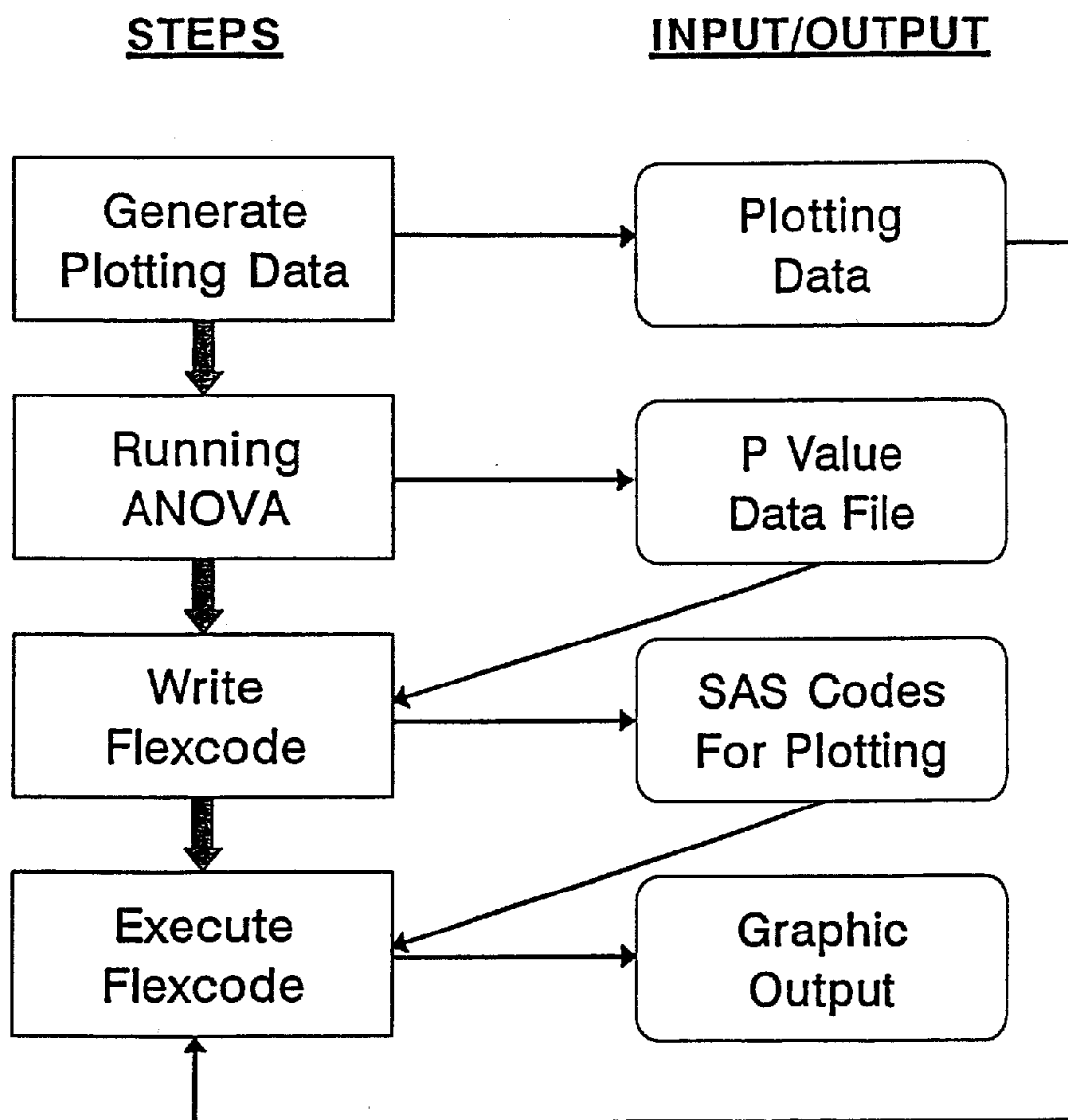
proc gplot data=e_g ;
  plot mean*e=g
  /legend=legend1 vaxis=axis1 vminor=1 haxis=axis2;
  where level=1 and item=8 ;
  title1 h=3.5 'Twoway Interaction of E and G ';
  title2 h=3.5 'Level=1, Item 8, P=0.043';

symbol1 v=square h=5 l=1 w=2 i=join c=black;
symbol2 v=diamond h=6 l=34 w=2 i=join c=black;

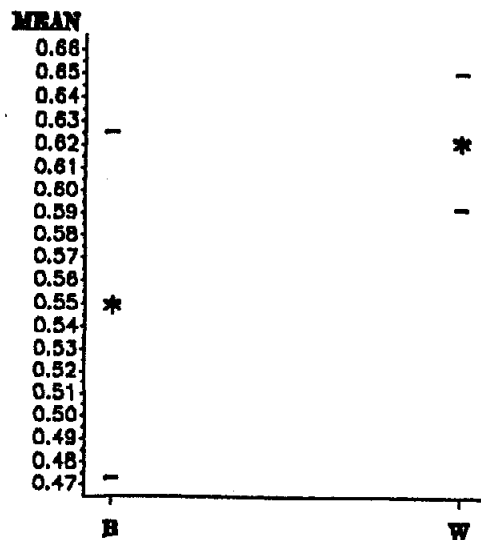
proc sort data=e_g_s;
  by g;

proc gplot data=e_g_s ;
  plot mean *e=s
  /legend=legend1 vaxis=axis1 vminor=1 haxis=axis2;
  by g;
  label g='00'x;
  where level=2 and item=1 ;
  title1 h=3.5 'Simple ES @ G ' ;
  title2 h=3.5 ' Level=2, Item 1, P=0.000 ' ;
```

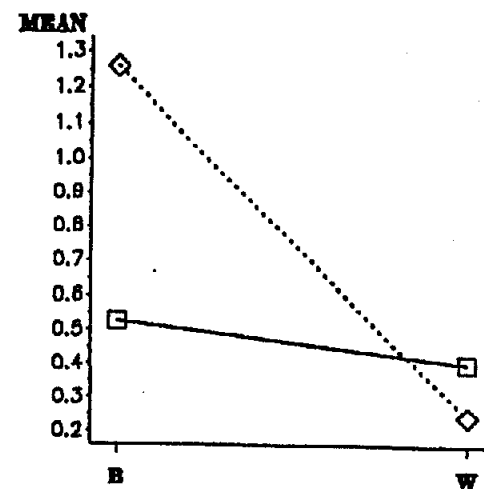
**Figure 1: An outline of the procedure.**



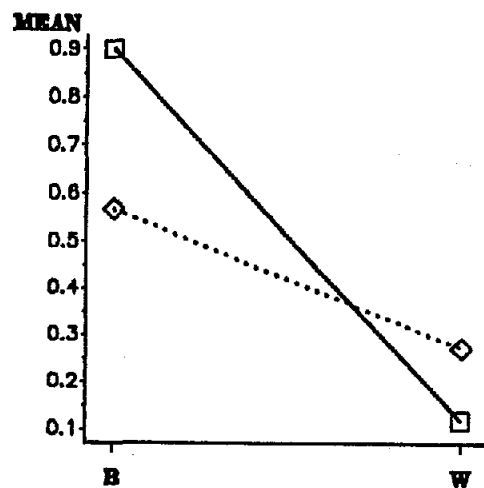
**Main Effects for E**  
Level 1, Item 3,  $P=0.021$



**Twoway Interaction of E and G**  
Level 1, Item 8,  $P=0.044$



**Simple ES @ F**  
Level 2, Item 1,  $P=0$



**Simple ES @ M**  
Level 2, Item 1,  $P=0$

