SAS PROC GLM and MIXED For Recovering Information in Augmented Designs

by

Russell D. Wolfinger,
SAS Institute R52
SAS Campus Drive
Cary, NC 27513
sasrdw@sas.com

Walter. T. Federer*
Warren Hall
Cornell University
Ithaca, NY 14853
wtf1@cornell.edu

Olga Cordero-Brana
Dept. Mathematics and Statistics
American University
Massachusetts Ave.
NW, Washington, D. C., 20016-8050
olgacb@american.edu.

*Corresponding author

Key words: computer codes, interblock information, interrow information, intercolumn informatiom, intervariety information, orthogonal polynomials, interregression information, mixed model, fixed effects.

BU-1378-M in the Technical Report Series of the Biometrics Unit, Cornell University, Ithaca, New York 14853, November, 1996

SAS PROC GLM and MIXED For Recovering Information in Augmented Designs

Abstract

The SAS GLM and MIXED procedures can be useful for experimenters desiring to analyze data from screening experiments using a member of the class of augmented experiment designs. Since application of the procedures is typically not straightforward for these designs, several programs of possible interest are described. We show how to recover interblocking and intervariety information when the blocking and varieties are random effects, how to arrange varietal responses in descending order, and a number of other options.

Abbreviations:

AED: augmented experiment design

ARCBD: augmented randomized complete block design

AIBD: augmented incomplete block design

ANOVA: analysis of variance

EBLUP: empirical best linear unbiased predictor

REML: restricted maximum likelihood

1. Introduction

The class of augmented designs was introduced by Federer (1956, 1961, 1991) as an alternative to the systematic arrangement of a single check variety in every kth plot. This latter method does not provide for an estimate of experimental error and is inefficient in that too much space is devoted to check plots (Yates, 1936). Also, the random nature of genotypes in the early stages of a selection program needs to be taken into account (Federer, 1996). Cullis *et al.* (1989) provide an analysis for the systematic check arrangement and consider the genotypes as random effects, but the over-abundance of check plots and the single check deficiencies remain. An augmented experiment design (AED) is obtained by selecting an

experiment design for checks and enlarging the blocks (rows and columns) to accommodate new genotypes (treatments) which usually only appear once in an experiment. Our purpose here is to present PROC GLM (SAS Institute Inc., 1989) and PROC MIXED (SAS Institute Inc., 1996) programs for obtaining the analysis for this class of designs in much the same manner as described by Federer and Wolfinger (1996). Describing codes for various statistical analyses make procedures readily available to an experimenter in order that more efficient statistical analyses and better use of resources may be made. Also, some other related program ideas are described.

2. Materials and Methods

The SAS software package with the PROC GLM (general linear models) and PROC MIXED (mixed model of random and fixed effects) was used to develop codes or programs for recovering interblocking and intervariety or intergenotype information associated with the random variables in the experimnet. Such variables as complete blocks, incomplete blocks, rows columns, and/or genotypes quality for consideration as random effects. Programs in PROC GLM consider all variables as fixed effects as is done in regression analyses. In this procedure some variables may be designated as random for the purpose of obtaining expected values of mean squares only. PROC MIXED uses REML (restricted maximum likelihood) solutions for the variance components for random effects but other options are available. Ordinary textbook analyses use analysis of variance (ANOVA) solutions for the variance components used for recovering intereffect information. Another procedure, PROC IML (interactive matrix language), was used to generate orthogonal polynomial regression coefficients for statistical analyses.

To develop codes for specific analyses, the desired statistical analysis was determined. Then, a matrix software package such as GAUSS was used to obtain the numerical values for an example. The example was constructed from known fixed effect parameters for all variables, allowing for a check on the GAUSS solutions obtained. Knowing the numerical values desired, various SAS commands were investigated to

obtain the desired results. When the commands giving the desired results were obtained, the program for the anlysis was finalized.

3. Results and Discussion

PROC GLM and PROC MIXED codes were developed for an augmented balanced incomplete block design using n = 6 new genotypes and c = 4 checks as an illustratrive example. A balanced incomplete block design for c = 4 checks in incomplete blocks of size two in r = 3 complete blocks or replicates was augmented by including one new treatment in each of the six incomplete blocks. The n +c = v = 10 entries need to be divided into a set (new) which represents the random set and a set (checks) which represent the entries considered to be fixed effects. The outputs for the PROC GLM and PROC MIXED codes are given in the Appendix. These same procedures plus PROC IML are used to develop a program for the anlysis of a 15 row by 12 column augmented design with n = 120 new and c = 2 checks replicated 30 times each. Since this row-column design is not connected (i.e., not all row, column, and entry effects have solutions under the usual restrictions that effects of a variable sum to zero), orthogonal polynomial regression functions of row and columns were used in the analysis. In addition, owing to the nature of spatial variation, interactions of the row and column regressions were needed to account for the particular type of variation encountered.

PROC GLM For AEDs With One-Way Blocking

With respect to an augmented randomized complete block (ARCBD) or an augmented incomplete block (AIBD) design, every blocked design is incomplete with respect to the new treatments. Hence, it is desirable to recover interblock information even for ARCBDs. However, for completeness, we begin by showing how to use SAS PROC GLM to obtain only intrablock and intravariety analyses for an AIBD as follows:

```
data augbibd;
  infile 'augbibd.dat';
  input yield rep block treat;
proc glm data=augbibd;
  class rep block treat;
  model yield = rep block(rep) treat / solution;
  random rep block(rep);
  lsmeans treat;
run;
```

The program starts with a SAS DATA step inputting a raw data file named "augbibd.dat" containing data from an AIBD with n new treatments and c check treatments. The input variables are YIELD (the response), REP (the replicate), BLOCK (the block), and TREAT (the treatment).

The DATA= option in the PROC GLM statement reads in the newly created SAS data set. The CLASS statement declares REP, BLOCK, and TREAT to be classification (qualitative) variables. The MODEL statement lists the dependent variable YIELD and the effects to be used in the analysis. Since REP, BLOCK, and TREAT are all classification variables, the effects involving them are constructed using 0-1 indicator variables.

Owing to the nature of PROC GLM's model parameterization, we assume the levels of TREAT are 1 to n for the n new genotypes and n +1, n + 2, ..., n + c for the c checks. The SOLUTION option in the MODEL statement prints out estimates of all of these levels, and since there is an overparameterization, PROC GLM sets the last treatment effect equal to zero. Therefore the new and the other check effects will all have the last treatment effect subtracted from them. The standard error listed with the solution is a standard error of a difference of the two effects, and the highest numbered check should be the one of most interest.

The RANDOM statement declares the REP and BLOCK(REP) effects to be random effects, leaving TREAT as the lone fixed effect in this analysis. In spite of being declared random, PROC GLM will still consider REP and BLOCK(REP) to be fixed effects during the model fit, but it will compute the expected mean squares for the replicate (complete block for checks) and blocks within replicates mean squares. Note that the sum of the new treatment effects are not required to sum to zero when using the constraint that the highest numbered treatment effect is equal to zero.

The LSMEANS statement computes estimated population marginal means for TREAT with equal weights applied to each of the treatment levels. The resulting estimates are the same as if using the constraint that the sum of all the treatment effects is zero. If it is desired to sort the lsmeans from the highest to the lowest, change the LSMEANS statement to the following:

```
lsmeans treat / out=lsmeans noprint;
```

and then add

```
proc sort data=lsmeans;
  by descending lsmean;
proc print;
run;
```

The NOPRINT option prevents the lsmeans from being printed during the PROC GLM invocation. Since most AEDs have large numbers of new treatments, the above ordering is a desirable feature for the experimenter who wishes to select the top performers and to discard poor performers.

To run an analysis on check yields only for n = 6 new treatments and c = 4 checks, add the following statement to the end of the DATA step:

```
if treat > 6 and treat < 11 then check = treat;
```

To obtain additional sums of squares, the following code can be useful:

```
data augbibd;
  infile 'augbibd.dat;
  input yield rep block treat;
  if (treat > 6) then new = 0;
  else new = 1:
  if (new) then treatn = 999;
  else treatn = treat;
proc glm data=augbibd;
  class rep block treat treatn;
  model yield = rep block(rep) treatn treat*new;
  random rep block(rep);
  lsmeans treatn;
run;
```

PROC MIXED for Recovering Interblocking Information

Although the preceding PROC GLM code can provide a fairly complete analysis of data from an augmented design, it can be a difficult chore sorting through the various sums of squares and constructing

193 appropriate tests. We therefore recommend PROC MIXED be used for most augmented design problems.
194 The output is much more straightforward and direct account is made of random effects.
195
196 Our first analysis using PROC MIXED considers both check and new treatments to be fixed effects and
197 replicates and blocks to be random effects:
198
199 data augbibd;
200 infile 'augbibd.dat';
201 input yield rep block treat;
202 proc mixed data = augbibd;
203 class rep block treat;
204 model yield = treat;
205 random rep block(rep);
206 lsmeans treat;
207 run;
208
209 Note the syntax for PROC MIXED is nearly identical to that of PROC GLM, with one important
210 exception: only fixed effects are listed in PROC MIXED's MODEL statement, whereas both fixed and
211 random effects are listed in PROC GLM's MODEL statement. As noted before, this is not really an
212 inconsistency because PROC GLM considers all effects to be fixed when it fits the linear model. On the
213 other hand, PROC MIXED handles random effects directly by estimating their variance components using
214 Gaussian restricted maximum likelihood.
215
216 The treatment means resulting from the LSMEANS statement are adjusted for interreplicate (for the new
217 treatments) and interblock information, and associated t-statistics take into account all estimated variance
218 components.
219

PROC MIXED for Recovering Both Interblocking and Intervariety Information

Since the SAS system is not designed to partition a variable such as TREAT into a set which is fixed (the checks) and a set which is random (the new treatments), it is necessary to construct some auxiliary variables in order to accomplish this. The following program considers the checks as fixed effects and the other effects as random, still assuming n = 6 new treatments and c = 4 checks:

```
data augbibd;
  infile 'augbibd.dat';
  input yield rep block treat;
  if (treat >  6) then new = 0;
  else new = 1;
  if (new) then treatn = 999;
  else treatn = treat;
proc mixed data=augbibd;
  class rep block treat treatn;
  model yield = treatn;
  random rep block(rep) treat*new / solution;
  lsmeans treatn;
  make 'solutionr' out=sr noprint;
run;
proc sort data=sr;
  by descending est;
proc print;
run;
```

The DATA step creates two new auxiliary variables: NEW and TREATN. NEW indicates whether or not a treatment is a new treatment, and is subsequently used to construct the random effect corresponding to the new treatments. TREATN equals TREAT for all of the check treatments but has a constant level for all of the new treatments. It is used as a fixed effect to model different means for each of the check treatments and a common mean for the new treatments. The new treatments are thus assumed to vary randomly about a common mean, and note this mean is free to fall anywhere in relation to the check means. The MODEL statement thus lists TREATN as the sole fixed effect, and the subsequent LSMEANS statement uses TREATN to construct mean estimates.

The RANDOM statement again lists REP and BLOCK(REP) as random effects along with a new one: TREAT*NEW. This last effect equals 0 for all of the check treatments (note that NEW is not a CLASS variable) and has a different level for all of the new treatments. The SOLUTION option in the RANDOM statement requests empirical best linear unbiased predictors (EBLUPs) of the random effects.

The MAKE statement is PROC MIXED's mechanism for creating output data sets, and the one listed here creates a data set named SR from the EBLUP table printed by the SOLUTION option in the RANDOM statement. The 'solutionr' string is the label for this table and is a necessary part of the MAKE statement. All such labels as well as detailed information on every statement can be found in the PROC MIXED documentation (SAS Institute Inc., 1996).

The final lines of the program sort and print the EBLUPs. Note that the EBLUPs for REP, BLOCK(REP), TREAT*NEW will all be intermixed in this printout, and one may wish to extract just those for TREAT*NEW in a different analysis. The sorted EBLUPs for TREAT*NEW provide a means for comparing the new treatments.

Other Augmented Designs

The above ideas for recovering interblock as well as intervariety information is easily extended to other augmented designs. The analysis described by Federer (1996) provides a useful example, and others can be found in Federer *et al.* (1975) and Federer and Raghavarao (1975). This example uses an input data file augmerc1.dat, and there are c = 2 checks repeated 30 times each in 15 rows and 12 columns with n = 120 new genotypes.

PROC GLM and PROC MIXED programs are now presented for obtaining Type III sums of squares, intrarow-column (fixed effects) least squares means, check treatment means adjusted for interrow and intercolumn information, and new treatment means adjusted for interrow, intercolumn, and intervariety information. A PROC GLM program for obtaining some relevant sums of squares is as follows:

```
data augmerc1;
  infile 'augmerc1.dat';
  input site col row treat gw c1 c2 c3 c4 r1 r2 r3 r4;
  if (treat > 120) then new = 0;
  else new = 1;
  if (new) then treatn = 999;
  else treatn = treat;
  ll = r1*c1;
  lq = r1*c2;
proc glm data=augmerc1;
  class row col treat treatn;
  model gw = r1 r2 r3 r4 c1 c2 c3 c4 ll lq treatn treat*new;
  random row col treat*new;
run;
```

The dependent variable GW is grain weight. Ri and Ci (i = 1, 2, 3, 4) are orthogonal polynomial regressions for row and column numbers. These variables can be created using the ORPOL function in SAS/IML. For example, the following program creates a data set OPN15 containing variables ROW and R1-R4. This data set can then be match-merged with the original data set.

```
proc iml;
  opn15 = orpol(1:15,4);
  opn15[,1] = (1:15)`;
  op15 = opn15;
  create opn15 from opn15[colname={'ROW' 'R1' 'R2' 'R3' 'R4'}];
  append from opn15;
  close opn15;
run;
```

The term columnname refers to the column in the creared data set and not to the column of the experiment design. In the preceding DATA step LL and LQ are created to represent interactions of row and column regressions. These are then specified along with the other variables in the MODEL statement of PROC GLM. This particular model is used here because the design is not connected. Other regression terms may be added to the model if deemed necessary to explain the experimental variation.

Note that the row, column, and new treatment effects are considered random. The RANDOM statement is used to obtain the expected values of mean squares in the event ANOVA solutions for the row, column, and new variance components are required.

The following code constructs the fixed effects means and arranges them in order from highest to lowest:

```
proc glm data=augmerc1;
```

```
328     class row col treat treatn;
329     model gw = r1 r2 r3 r4 c1 c2 c3 c4 ll lq treat;
330     lsmeans treat / out = lsmeans noprint;
331   run;
332   proc sort data=lsmeans;
333     by descending lsmean;
334   proc print;
335   run;
336
```

337 While the preceding PROC GLM code can be used to obtain various partitions of the sums of squares, a
338 more straightforward analysis can be obtained with PROC MIXED. This analysis adjusts the check
339 means for interrow and intercolumn information and adjusts the new effects for interrow, intercolumn,
340 and intervariety information:

```
341
342   proc mixed data=augmerc1;
343     class row col treat treatn;
344     model gw = r1 r2 r3 r4 c1 c2 c3 c4 ll lq treatn / solution;
345     random r1 r2 r3 r4 c1 c2 c3 c4 ll lq treat*new / solution;
346     lsmeans treatn;
347     make 'solutionr' out=sr noprint;
348   run;
349   proc sort data=sr;
350     by descending est;
351   proc print;
352   run;
353
```

If there were no row and column regression interactions, the Ri and Ci in the random statement could be replaced with ROW and COL as REML makes use of the normal distribution theory and the design need not be connected for the PROC MIXED procedure. Using the row and column designation for random corrects for all row and coumn effects and not just the Ri and Ci used in the regression model. Note that TREATN is used in the MODEL statement, and it has a distinct level for all of the check treatments and a common level (999) for all of the new treatments. The effect TREAT*NEW in the RANDOM statement models all of the new treatments as random effects varying about the common mean modeled by the 999 level of TREATN. For some computer set-ups, it may be necessary to use the command _EST_ in place of EST.

A report describing the use of the above programs has been prepared by Federer and Wolfinger (1996a). A small numerical example has been used and the computer outputs of the programs have been annotated with descriptions of the results.

4. Conclusions

Present software literature is inadequate for experimenters to obtain programs for the analyses described herein. Considerable computer and package expertise and several trial and error runs were required to obtain the final programs. These programs are in a readily usable form for experimenters who desire statistical analyses of augmented experiment designs and to recover interblock, interrow, intercolumn, interregression, and/or intervariety information. Recovery of the information associated with the random effects leads to more efficient analyses of experimental data, and hence more efficient use of experimental resources.

Acknowledgement

Appreciation is expressed for the constructive comments of an associate editor and referees. These were useful in clarifying presentation of results.

381 References

382

383 Cullis, B. R., J. L. Warwick, J. A. Fisher, and B. J. Read (1989). A new procedure for the analysis of
384 early generation variety trials. Applied Statistics 38:361-375.

385

386 Federer, W. T. (1956). Augmented (or hoonuiaku) designs. Hawaiian Planters' Record LV(2):191-208.

387

388 Federer, W. T. (1961). Augmented designs with one-way elimination of heterogeneity. Biometrics
389 17:447-473.

390

391 Federer, W. T. (1991). Statistics and Society, 2nd edition. Marcel Dekker, Inc., New York, Section 7.11.

392

393 Federer, W. T. (1996). Recovery of interblock, intergradient, and intervariety information in incomplete
394 block and lattice rectangle designed experiments. Biometrics (in press).

395

396 Federer, W. T. , R. C. Nair, and D. Raghavarao (1975). Some augmented row-column designs.
397 Biometrics 31:361-373.

398

399 Federer, W. T. and D. Raghavarao (1975). On augmented designs. Biometrics 31:29-35.

400

401 Federer, W.T. and R. D. Wolfinger (1996). SAS PROC GLM and PROC MIXED Code for Recovering
402 Inter-Factor Information. BU-1330-M in the Technical report Series of the Biometrics Unit, Cornell
403 University, Ithaca, New York.

404

405 Federer, W. T. and R. D. Wolfinger (1996a). GAUSS and SAS for recovering interblock and intervariety
406 information. BU-1384-M in the Technical Report Series of the Biometrics Unit, Cornell University,
407 Ithaca, New York.

SAS Institute Inc. (1989). SAS/STAT Users Guide, Version 6, Fourth Edition, Volume 2. SAS Institute Inc., Cary, NC.

SAS Institute Inc. (1996). SAS/STAT Software: Changes and Enhancements through Release 6.11. SAS Institute Inc., Cary, NC.

Yates, F. (1936). A new method of arranging variety trials involving a large number of varieties. Journal of Agricultural Science 26:424-455.