Paper ST04

Application of the ARRAY statement to create ESTIMATE and CONTRAST statements in statistical PROCEDURES: a case study.

Roel Straetemans, Janssen Pharmaceutical, Beerse, Belgium
Franky De Cooman, GBSC, Gent, Belgium

ABSTRACT

CONTRAST and ESTIMATE statements are important tools when performing statistical analyses. For a statistician the creation of the correct code can however be cumbersome when dealing with many variables and/or different levels within variables. This paper will illustrate how the ARRAY statement can be used to circumvent these difficulties by letting SAS create the correct and necessary code.

INTRODUCTION

In SAS the CONTRAST and ESTIMATE statements are commonly used to test hypotheses and estimate linear combination of parameter values within different procedures, e.g. MIXED, GENMOD, GLM, . The complete technical details can be found in the manual for each of the procedures but in essence a matrix of coefficients needs to be written with the dimension of the matrix depending on the number of variables in the model and, if a variable is categorical, also on the number of levels present. As a consequence by changing the model (for example during model selection procedures or including extra levels of a certain variable) the CONTRAST and ESTIMATE statements typically need to be manually adjusted. This can be cumbersome work and creates room for error.

At Janssen Pharmaceutical an automated statistical analyses procedure needed to be developed for the Delayed Non-Matching To Position experiment (DNMTP) [1]. The statistical analysis involved a generalized estimation equation (GEE) approach [2-4] using GENMOD. More details on the experiment and on the statistical analysis will be given in the remainder of the paper. Essential in the automatisation program was the possibility to calculate certain CONTRAST and ESTIMATE statements. Since both the number of variables included in the final model as well as number of levels of the main variables could change from experiment to experiment the SAS program needed to be able to write it's own matrices. An easy and elegant solution to this problem was obtained by making use of the ARRAY statement in SAS. The main purpose of this paper is to illustrate how this was done. Because of the well spread use of the CONTRAST and ESTIMATE statement in statistical analyses we think this approach could be of interest to many users since it avoids manual generation of the matrices.

The rest of the paper is organized as follows. The next section contains a short description of the DNMTP experiment, followed by a section covering more details on the statistical analyses performed as well as the problem on hand concerning programmation of the necessary ESTIMATE and CONTRAST statements. Finally, the application of the ARRAY statement to implement automated generation of the necessary CONTRAST and ESTIMATE statements is fully explained in and illustrated for the ESTIMATE statement in the final section. Extension for the CONTRAST statement is straightforward.

DNMTP EXPERIMENT

BACKGROUND

The DNMTP experiment is a standard experiment to investigate drug effects on spatial recognition memory in rats. In summary a rat is trained to follow an elaborate procedure, which when successfully completed results in a reward in the form of a food pellet. The procedure consists of three tasks in a fixed order. First a sample lever needs to be pushed, next a nose poke needs to be performed within a delay period (typically there are 6 different delay periods within one experiment) and finally a choice lever needs to be pushed. This procedure of the three tasks in DNMTP terminology is called a 'trial'. A more elaborate technical description can be found in the literature [1]. After drug

administration it is investigated how successful the animals were able to remember how to successfully complete the procedure.

EXPERIMENTAL DESIGN

The study design of the DNMTP experiment is a cross-over setting [5]. Briefly, each animal receives each dose in a sequence. When 3 or more treatments are investigated, as typically in a DNMTP experiment, multiple sequences are possible. In our case a Latin Square design is applied and rats are randomized to each of the possible sequence within the Latin Square. During a treatment period each of the six delay periods is investigated (in a random order) and an animal typically goes through 10 'trials' for each combination of delay period and dose level. Thus if for example 3 treatments are investigated, each animal goes through 180 trials, divided in 60 trials for each treatment period, involving 10 trials for each delay.

GEE ANALYSIS

The primary outcome of interest is the success probability of a rat completing the 'trial', resulting in a reward in the form of food or not. The response variable is considered a binomial variable having two levels: success or failure. The typical analysis for this type of response variable is a logistic regression. However since the data from this experiment are clustered (multiple observations on each animal), the multivariate structure of the data needs to be taken into account. This can typically be done using a GEE analysis.

POSSIBLE STATISTICAL MODELS

Remember how the effect of two variables on the response was of importance, delay and dose. As a result two types of structural models are possible, one with an interaction effect between these two variables and the other with only main effects. The choice for the final model is based on the Akaike [6] information criterion (AIC). The specifics of this procedure are of no importance for this paper, however important is to realize the impact on the ESTIMATE and CONTRAST statement.

For both models, the user requested an estimate of the effect for each dose delay combination. This could be obtained using the ESTIMATE statement. Moreover if an interaction model was found to be the appropriate model, the user wanted to investigate which of the dose effects we're important at which delays. For this purpose CONTRAST statements were necessary.

ILLUSTRATION OF THE PROBLEM

To illustrate the essence of the problem at hand, SAS code is presented, in CODE 1, for an interaction model involving 3 doses and 6 delays.

The two ESTIMATE statements show the necessary matrices to estimate the effect of the first dose at the first delay and at the second delay. The two CONTRAST statements illustrate the necessary matrices to investigate at which delay a dose effect is present.

The precise statistical details behind these models are not necessary to understand and are not important, however it is clear that subtle differences in the matrices are necessary to obtain the desired result. Also above is just a fraction of the code displayed. Remember that when having 3 doses and 6 delays we need 18 ESTIMATE and 18 CONTRAST statements. When 4 doses and 6 delays are involved this expands to 24 ESTIMATE and 24 CONTRAST statements. Moreover when the final correct model is the main effects model (see CODE 2) the matrices needed in the ESTIMATE statement change.

The program needed to be able to handle experiments where the number of dose levels ranged from 3 to 6 and the number of delay levels possibly could ranged from 3 to 8. This caused quite a challenge to code the ESTIMATE and CONTRAST statement. Therefore we wanted to develop a program that could determine itself which ESTIMATE and CONTRAST statements are necessary and subsequently create the necessary matrices.

```
PROC GENMOD DATA=master2 DESCENDING ORDER=formatted;
      CLASS subject period dose carry delay;
      MODEL y = delay dose delay*dose / DIST=bin type3 ;
      REPEATED subject=subject / corr=&variance covb corrw modelse;
      OUTPUT OUT = estimates ;
      ESTIMATE  'Dose1 at delay1'
            intercept   1
            dose        1 0 0
            delay       1 0 0 0 0 0
            dose*delay  1 0 0 0 0 0       0 0 0 0 0 0       0 0 0 0 0 0;

      ESTIMATE  'Dose1 at delay2'
            intercept   1
            dose        1 0 0
            delay       0 1 0 0 0 0
            dose*delay  0 1 0 0 0 0       0 0 0 0 0 0       0 0 0 0 0 0;

      CONTRAST 'Dose effect at Delay1'
            dose        1 0 -1
            dose*delay  1 0 0 0 0 0       0 0 0 0 0 0       -1 0 0 0 0 0,
            dose        0 1 -1
            dose*delay  0 0 0 0 0 0       1 0 0 0 0 0       -1 0 0 0 0 0 ;

      CONTRAST 'Dose effect at Delay2'
            dose        1 0 -1
            dose*delay  0 1 0 0 0 0       0 0 0 0 0 0       0 -1 0 0 0 0,
            dose        0 1 -1
            dose*delay  0 0 0 0 0 0       0 1 0 0 0 0       0 -1 0 0 0 0 ;

RUN;
```

CODE 1: Interaction model

```
PROC GENMOD DATA=master2 DESCENDING ORDER=formatted ;
      CLASS subject period dose carry delay;
      MODEL y = delay dose delay*dose / DIST=bin type3 ;
      REPEATED subject=subject / corr=&variance covb corrw modelse;
      OUTPUT OUT = estimates ;
      ESTIMATE  'Dose1 at delay1'
            intercept   1
            dose        1 0 0
            delay       1 0 0 0 0 0;

      ESTIMATE  'Dose1 at delay2'
            intercept   1
            dose        1 0 0
            delay       0 1 0 0 0 0;

RUN;
```

CODE 2: Main Effects model

ALL ROUND SAS PROGRAM

The following program is a sketch of what the code-generating program should look like;

```
DATA master(DROP = seed);
    seed = 1;
    DO dose = 0, 0.6, 1, 5;
          DO delay = 0 TO 20 BY 5;
                DO subject_id = 1 TO 10;
                      chocor = RANBIN(seed, 1, 0.5);
                      OUTPUT;
                END;
          END;
    END;
RUN;


proc sql;
create table doses as
select distinct dose
from master
order by dose;
quit;

proc sql;
create table delays as
select distinct delay
from master
order by delay;
quit;

DATA doses;
    SET doses END =eof;
    idose + 1;
    IF eof THEN CALL SYMPUT('ndose', TRIM(LEFT(PUT(idose, best12.))));
RUN;

DATA delays;
    SET delays END = eof;
    idelay + 1;
    IF eof THEN DO;
          CALL SYMPUT('ndelay', TRIM(LEFT(PUT(idelay, best12.))));
          CALL SYMPUT('ndd', TRIM(LEFT(PUT(idelay * &ndose, best12.))));
    END;
RUN;


PROC SQL NOPRINT;
    CREATE TABLE dosedelay AS
    SELECT DISTINCT *
    FROM doses o,
         delays e
    ;
QUIT;

DATA estimate;
    SET dosedelay;
    LENGTH dosecod delaycod dosedelaycod $200;
    ARRAY doseA {*} dose1 - dose&ndose;
    ARRAY delayA {*} delay1 - delay&ndelay;
```

```
    ARRAY dosedelayA {&ndose, &ndelay} dd1 - dd&ndd;
    ARRAY dosedelayB {*} dd1 - dd&ndd;
    DO i = 1 TO DIM(doseA);
          doseA{i} = 0;
    END;
    DO i = 1 TO DIM(delayA);
          delayA{i} = 0;
    END;
    DO i = 1 TO DIM(dosedelayB);
          dosedelayB{i} = 0;
    END;
    doseA{idose} = 1;
    delayA{idelay} = 1;
    dosedelayA{idose, idelay} = 1;
    DO i = 1 TO DIM(doseA);
          dosecod = TRIM(LEFT(dosecod)) !! ' ' !! TRIM(LEFT(PUT(doseA{i}, 1.)));
    END;
    DO i = 1 TO DIM(delayA);
          delaycod = TRIM(LEFT(delaycod)) !! ' ' !! TRIM(LEFT(PUT(delayA{i}, 1.)));
    END;
    DO i = 1 TO DIM(dosedelayA, 1);
          DO j = 1 TO DIM(dosedelayA, 2);
                dosedelaycod = TRIM(LEFT(dosedelaycod)) !! ' ' !!
TRIM(LEFT(PUT(dosedelayA{i,j}, 1.)));
          END;
          dosedelaycod = TRIM(LEFT(dosedelaycod)) !! '%';
    END;
    dosedelaycod = TRANWRD(dosedelaycod, '%', '    ');
RUN;



DATA oneobs;
    obs = 1;
    OUTPUT;
RUN;


PROC SORT DATA = master;
    BY subject_id /*session_id*/;
RUN;


%LET respvar = chocor;
%LET params = dose delay delay*dose;
%LET variance = exch;

FILENAME GEE1 CATALOG 'work.source.GEE1.source';


DATA code11;
    LENGTH code $200;
    SET oneobs;
    code = 'PROC GENMOD DATA=master DESCENDING ORDER=formatted ;'; OUTPUT;
    code = "      CLASS subject_id dose delay;";OUTPUT;
    code = "    MODEL &RESPVAR = &params / DIST=bin type3 ;";OUTPUT;
    code = "    REPEATED subject=subject_id / corr=&variance covb corrw modelse
;*CONVERGE=0.01;";OUTPUT;
    code = "      OUTPUT OUT = estimates ;";
RUN;

DATA code12;
```

```
    LENGTH code $200;
    SET estimate;
    code = "    ESTIMATE 'Dose " !! TRIM(LEFT(PUT(dose, best12.))) !! ' @ Delay ' !!
TRIM(LEFT(PUT(delay, best12.))) !!"'"; OUTPUT;
    code = "           intercept 1";OUTPUT;
    code = "           dose " !!dosecod; OUTPUT;
    code = "           delay " !! delaycod;;OUTPUT;
    code = "           dose*delay " !! dosedelaycod;OUTPUT;
    code = "           ;          ";OUTPUT;
RUN;



DATA code13;
    LENGTH code $200;
    SET oneobs;
    code = "   LSMEANS dose*delay / CL diff;"; OUTPUT;
    code = "   ODS output GEEEmpPEst=paramestimates Type3=type3 LSMEANS=lsmeans ";
OUTPUT;
    code = "            LSMeanDiffs=lsmeandiff Estimates=pointestimate;"; OUTPUT;
    code = "   QUIT;"; OUTPUT;
RUN;

DATA _NULL_;
    SET code11 code12 code13;
    FILE GEE1;
    PUT code;
RUN;

%INC GEE1;

if mix(var1, var2) > 0 then do;
```

This then generates the following piece of code found in *GEE1*

```
PROC GENMOD DATA=master DESCENDING ORDER=formatted ;
CLASS subject_id dose delay;
MODEL chocor = dose delay delay*dose / DIST=bin type3 ;
REPEATED subject=subject_id / corr=exch covb corrw modelse ;*CONVERGE=0.01;
ESTIMATE 'Dose 0 @ Delay 0'
intercept 1
dose 1 0 0 0
delay 1 0 0 0 0
dose*delay 1 0 0 0 0     0 0 0 0 0     0 0 0 0 0     0 0 0 0 0
;
ESTIMATE 'Dose 0 @ Delay 5'
intercept 1
dose 1 0 0 0
delay 0 1 0 0 0
dose*delay 0 1 0 0 0     0 0 0 0 0     0 0 0 0 0     0 0 0 0 0
;
ESTIMATE 'Dose 0 @ Delay 10'
intercept 1
dose 1 0 0 0
delay 0 0 1 0 0
dose*delay 0 0 1 0 0     0 0 0 0 0     0 0 0 0 0     0 0 0 0 0
;
ESTIMATE 'Dose 0 @ Delay 15'
intercept 1
```

```
dose 1 0 0 0
delay 0 0 0 1 0
dose*delay 0 0 0 1 0     0 0 0 0 0     0 0 0 0 0     0 0 0 0 0
;
ESTIMATE 'Dose 0 @ Delay 20'
intercept 1
dose 1 0 0 0
delay 0 0 0 0 1
dose*delay 0 0 0 0 1     0 0 0 0 0     0 0 0 0 0     0 0 0 0 0
;
ESTIMATE 'Dose 0.6 @ Delay 0'
intercept 1
dose 0 1 0 0
delay 1 0 0 0 0
dose*delay 0 0 0 0 0     1 0 0 0 0     0 0 0 0 0     0 0 0 0 0
;
ESTIMATE 'Dose 0.6 @ Delay 5'
intercept 1
dose 0 1 0 0
delay 0 1 0 0 0
dose*delay 0 0 0 0 0     0 1 0 0 0     0 0 0 0 0     0 0 0 0 0
;
ESTIMATE 'Dose 0.6 @ Delay 10'
intercept 1
dose 0 1 0 0
delay 0 0 1 0 0
dose*delay 0 0 0 0 0     0 0 1 0 0     0 0 0 0 0     0 0 0 0 0
;
ESTIMATE 'Dose 0.6 @ Delay 15'
intercept 1
dose 0 1 0 0
delay 0 0 0 1 0
dose*delay 0 0 0 0 0     0 0 0 1 0     0 0 0 0 0     0 0 0 0 0
;
ESTIMATE 'Dose 0.6 @ Delay 20'
intercept 1
dose 0 1 0 0
delay 0 0 0 0 1
dose*delay 0 0 0 0 0     0 0 0 0 1     0 0 0 0 0     0 0 0 0 0
;
ESTIMATE 'Dose 1 @ Delay 0'
intercept 1
dose 0 0 1 0
delay 1 0 0 0 0
dose*delay 0 0 0 0 0     0 0 0 0 0     1 0 0 0 0     0 0 0 0 0
;
ESTIMATE 'Dose 1 @ Delay 5'
intercept 1
dose 0 0 1 0
delay 0 1 0 0 0
dose*delay 0 0 0 0 0     0 0 0 0 0     0 1 0 0 0     0 0 0 0 0
;
ESTIMATE 'Dose 1 @ Delay 10'
intercept 1
dose 0 0 1 0
delay 0 0 1 0 0
```

```
dose*delay 0 0 0 0 0     0 0 0 0 0     0 0 1 0 0     0 0 0 0 0
;
ESTIMATE 'Dose 1 @ Delay 15'
intercept 1
dose 0 0 1 0
delay 0 0 0 1 0
dose*delay 0 0 0 0 0     0 0 0 0 0     0 0 0 1 0     0 0 0 0 0
;
ESTIMATE 'Dose 1 @ Delay 20'
intercept 1
dose 0 0 1 0
delay 0 0 0 0 1
dose*delay 0 0 0 0 0     0 0 0 0 0     0 0 0 0 1     0 0 0 0 0
;
ESTIMATE 'Dose 5 @ Delay 0'
intercept 1
dose 0 0 0 1
delay 1 0 0 0 0
dose*delay 0 0 0 0 0     0 0 0 0 0     0 0 0 0 0     1 0 0 0 0
;
ESTIMATE 'Dose 5 @ Delay 5'
intercept 1
dose 0 0 0 1
delay 0 1 0 0 0
dose*delay 0 0 0 0 0     0 0 0 0 0     0 0 0 0 0     0 1 0 0 0
;
ESTIMATE 'Dose 5 @ Delay 10'
intercept 1
dose 0 0 0 1
delay 0 0 1 0 0
dose*delay 0 0 0 0 0     0 0 0 0 0     0 0 0 0 0     0 0 1 0 0
;
ESTIMATE 'Dose 5 @ Delay 15'
intercept 1
dose 0 0 0 1
delay 0 0 0 1 0
dose*delay 0 0 0 0 0     0 0 0 0 0     0 0 0 0 0     0 0 0 1 0
;
ESTIMATE 'Dose 5 @ Delay 20'
intercept 1
dose 0 0 0 1
delay 0 0 0 0 1
dose*delay 0 0 0 0 0     0 0 0 0 0     0 0 0 0 0     0 0 0 0 1
;
LSMEANS dose*delay / CL diff;
ODS output GEEEmpPEst=paramestimates Type3=type3 LSMEANS=lsmeans
LSMeanDiffs=lsmeandiff Estimates=pointestimate;
QUIT;
```

In the next paragraphs, we will explain the main technologies used to obtain the resulting program

CREATION OF DOSES, DELAYS, DOSEDELAY DATASETS

These datasets will hold the unique values of the variables to be modeled.

CREATION OF ESTIMATE DATASET

Using the ARRAY statement, this dataset will hold for each of the combinations, the corresponding flags (0 and 1's) to construct the estimate statement: *dosecod* for the *dose* variable, *delaycod* for the *delay* variable and *dosedelaycod* for the interaction.

CREATION OF GEE1 CODE FILE

With a filename statement, the GEE1 file is allocated, and three datasets are constructed. Each of the three datasets (code11, code12, code13) will contain a variable named *code*, which will contain the actual sas code to be processed.

code11

Holds the general code, to start up the GENMOD procedure, and to indicate the model

code12

This dataset is a placeholder for all the estimate statements, and is based on the *estimate* dataset.

code13

This table contains the code to finalize the GENMOD procedure

ASSEMBLY AND EXECUTION OF CODE

With a DATA _NULL_ the three coding datasets are assembled and filed into GEE1. With an include statement, the code is then executed.

CONCLUSION

An approach was introduced where the ARRAY statement is used to create correct code for ESTIMATE and CONTRAST statements, two important tools in performing statistical analyses. Manual coding of these statements can be cumbersome, time consuming and mistakes are easily made. These problems can however be overcome by using the ARRAY statement. The approach made creation of an automated analysis procedure for the DNMTP experiment possible.

REFERENCES

[1] Estapé, N. and Steckler, T. (2002**)** Cholinergic blockade impairs performance in operant DNMTP in two inbred strains of mice. *Pharmacol. Biochem. Behav.,* **72**, 319-334 .

[2] Diggle, P.J., Heagerty, P.J., Liang, K.-Y., and Zeger, S.L. (2002) *Analysis of Longitudinal Data (2nd edition)*. Oxford Science Publications. Oxford: Clarendon Press.

[3] Liang, K.-Y. and Zeger, S.L. (1986) Longitudinal data analysis using generalized linear models. *Biometrika*, **73**, 13-22.

[4] Zeger, S.L. and Liang, K.-Y. (1986) Longitudinal data analysis for discrete and continuous outcomes. *Biometrics*, **42**, 121-130.

[5] Jones, B. and Kenward, M.G. (1989) *Design and Analysis of Cross-Over Trials*. London: Chapman & Hall.

[6] Akaike, H. (1973) Information theory and an extension of the maximum likelihood principle. In Petrov, B.N. and Csàki, F. (eds) *Second International Symposium on Inference Theory*, Budapest: Akadémiai Kiadó, pp.267-281.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Roel Straetemans
Janssen Pharmaceutica

Turnhoutseweg 30
B-2340 Beerse
België

Work Phone: +32 14 / 60 27 44
Fax: 014 / 60 58 02
Email: RSTRAETE@PRDBE.jnj.com.be

Franky De Cooman

GBSC

Kerkstraat 108
9050 Gentbrugge

Work Phone: +32 476/26 10 25
Email: franky.de.cooman@gbsc.be