The previous paper requires some clarification about the use of the CATMOD procedure to fit the given market-share models. It should be pointed out that there is no R= option on the MODEL statement in CATMOD. The author is apparently referring to the feature in which the design matrix can be specified directly by the user. This feature allows CATMOD to fit a much wider class of models than it could otherwise fit. The design matrix may be large, but it is easily generated by a DATA step. This, combined with 1 PROC PRINT statement, and 4 CATMOD statements, enables the user to fit the simple-effects model, the differential-effects model, and other types of conditional logit models with very little effort.

As an example of how to fit these models with CATMOD, consider the following hypothetical data containing prices and sales for 4 different brands in 6 weeks.

```plaintext
/* data design; set prices; n=4; */
array inter{*} inter1-inter4;
array price{*} price1-price4;
array apricio{*} apricelo-price4;
keep inter1-inter3 apricio1-apricio3 comma;
inter(i) = 1; do i = 2 to n; inter(i) = 0; end;
apricio1 = price1-price(n);
do i = 2 to n; apricio(i) = 0; end;
do i = 1 to (n-1); comma = ',
output;
inter(i) = 0; inter(i+1) = 1;
apricio3 = 0;
apricio(i+1) = price(i+1) - price(n); end;
```

Again, the matrix is copied and merged with the other CATMOD statements, and the last line of the MODEL statement is changed to:

```plaintext
(1 2 3 = 'Intercept', 4 = 'Price sensitivity');
```

The output includes a goodness-of-fit test, parameter estimates, and the corresponding test statistics. The goodness-of-fit test indicates that the simple-effects model does not fit these data.

Fitting the differential-effects model requires a new design matrix that is a simple variation of the old one. It can be generated by a DATA step that is similar to the previous one:

```plaintext
/* data design; set prices; n=4; */
array inter{*} inter1-inter4;
array price{*} price1-price4;
array apricio{*} apricio1-apricio3;
keep inter1-inter3 apricio1-apricio3 comma;
inter(i) = 1; do i = 2 to n; inter(i) = 0; end;
apricio1 = price1-price(n);
do i = 2 to n; apricio(i) = 0; end;
do i = 1 to (n-1); comma = ',
output;
inter(i) = 0; inter(i+1) = 1;
apricio3 = 0;
apricio(i+1) = price(i+1) - price(n); end;
```

This model fits the data well, and it shows that price sensitivity is highly significant.

The extended (or cross-competitive) model is easily fit with the CATMOD procedure. The required CATMOD statements are similar to those contained in the previous paper except that, as above, one should use the adjusted prices since each logit is a comparison of the first and the fourth brands. The correct specification is:

```plaintext
/* data design; set prices; n=4; */
array inter{*} inter1-inter4;
array price{*} price1-price4;
keep inter1-inter3 apricio1-apricio3 comma;
inter(i) = 1; do i = 2 to n; inter(i) = 0; end;
apricio1 = price1-price(n);
do i = 2 to n; apricio(i) = 0; end;
do i = 1 to (n-1); comma = ',
output;
inter(i) = 0; inter(i+1) = 1;
apricio3 = 0;
apricio(i+1) = price(i+1) - price(n); end;
```

With respect to CATMOD's inability to calculate elasticities, it should be noted that CATMOD is a modeling procedure, not a scoring procedure. CATMOD produces output data sets containing observed and predicted cell frequencies, as well as estimated model parameters. These components and the design matrix can be combined in a straightforward way with user-input values of the relevant price points for the purpose of computing elasticities or other scores.