References

CHEATHAM, T. E., Jr., and SATTLEY, K. (1964). Syntax-Directed Compiling, presented at the Spring Joint Computer Conference April 21–23, 1964. (Computer Associates document # CA-64-1-R.)

Computer Associates (1963). Compiler Generator Systems Program Descriptions. (Computer Associates document # CA-63-4-SD, 1 July, 1963.)

IRONS, E. T. (1961). A Syntax-Directed Compiler for ALGOL 60, *Communications of the ACM*, Vol. 4, pp. 51–55.

WARSHALL, S., and SHAPIRO, R. M. (1964). A General-Purpose Table-Driven Compiler, presented at the Spring Joint Computer Conference, April 21–23, 1964. (Computer Associates document # CA-63-4-R.)

Errata

'Seasonal adjustment and forecasting in the presence of a trend', by R. W. Hiorns.

The ALGOL procedure presented at the end of the above paper, which appeared in Vol. 10, p. 143, is incorrect. The correct procedure is as follows.

procedure *season* (*y*, *m*, *f*, *n*, *N*, *a*, *e*, *w*, *P*1, *P*2, *pred*, *pe*);
value *y*, *m*, *f*, *n*, *N*, *P*1, *P*2; **array** *y*, *f*, *a*, *e*, *pred*, *pe*;
integer array *n*; **integer** *m*, *N*, *P*1, *P*2; **real** *w*;
comment *This procedure calculates, simultaneously, estimates of seasonal and trend constants by multiple linear regression and provides forecasts and errors for a specified period. The model used has the demand variable represented by a trend component, seasonal component and random component combined additively. The trend term is assumed to consist of a single function whose values are supplied to the procedure in an array* f[1 : N] *where N is the number of observed values of the demand variable, also supplied, in an array* y[1 : N]. *These values correspond to consecutive time periods, there being m seasons in a year, represented by m seasonal constants, but N need not be a multiple of m. N must satisfy* $N \geqslant m + 2$. *The number of observed values for each season must be supplied in the array* n[1 : m].

Estimates are left by the procedure as follows: the trend constant in a[0] *and the m seasonal constants in the remainder of the array* a[0 : m]. *Standard errors for the constants are in the array* e[0 : m]. *The residual variance estimate is in w.*

Forecasts (or predictions) are made for consecutive time periods from P1 to P2. These are left in pred[P1 : P2] *and their standard errors in* pe[P1 : P2].

It should be noted that if the values of P1 and P2 do not both lie within the range 1 to N, then the array f will require new bounds, the lesser of 1 and P1 and the greater of N and P2, respectively. In any case, values must be supplied to the whole of the array f before activation;

```
begin integer i, j, k; real F, YF, p, q; array T, Y[1 : m];
  YF:=F:=p:=q:=0;
     for i:=1 step 1 until m do
       begin
       k:=i; T[i]:=Y[i]:=0;
       for j:=1 step 1 until n[i] do
         begin
         F:=F+f[k]↑2; T[i]:=T[i]+f[k];
         Y[i]:=Y[i]+y[k]; YF:=YF+y[k]×f[k];
         k:=k+m
         end;
       p:=p−T[i]↑2/n[i]; q:=q−T[i]×Y[i]/n[i]
       end;
   p:=p+F; q:=q+YF;
estimates:
   a[0]:=q/p;
   for i:=1 step 1 until m do
   a[i]:=(Y[i]−a[0]×T[i])/n[i];
sumsquares:
   w:=0;
   for i:=1 step 1 until m do
     begin
     k:=i;
     for j:=1 step 1 until n[i] do
       begin
       w:=w+(y[k]—a[i]−a[0]×f[k])↑2; k:=k+m
       end
     end;
   w:=w/(N—m—1); e[0]:=sqrt(w/p);
   for i:=1 step 1 until m do
   e[i]:=sqrt((1+T[i]↑2/(p×n[i]))/n[i]×w);
predictions:
   for k:=P1 step 1 until P2 do
     begin
     i:=k−m×((k−1)÷m);
     if i⩽0 then i:=i+m;
     pred[k]:=a[i]+a[0]×f[k];
     pe[k]:=w×(1/n[i]+(T[i]/n[i]−f[k])↑2/p);
     pe[k]:=sqrt (pe[k])
     end
   end of season
```