

MACRO-EXPRESSION DE LA REGRESSION NON LINEAIRE

Sur la base du tableau de données *filin*, supposé contenir les observations temporelles en colonne et les variables en ligne (après avoir, par exemple, utilisé la procédure PROC TRANSPOSE), la macro-expression *voisin* réalise, grâce aux arguments présents dans le tableau ci-dessous, une régression non linéaire qu'on appelle aussi analyse de voisinage.

| Argument | Signification |
|-----------------|---|
| Filin | Nom du tableau (Data) d'entrée Sas . |
| Filout | Nom du tableau (Data) de sortie Sas qui contient l'identifiant <i>ident</i> , le numéro d'ordre du voisin, la distance euclidienne au voisin et, à la ligne numéro <i>n</i> , la distance au centre de gravité des <i>n</i> premiers voisins. |
| ident et valeur | Chaînes de caractères d'identification des observations. |
| Def et fin | Le voisinage est calculé pour la période allant de COL&deb à COL&fin. |

Quand *ident* est égal à *valeur*, la macro-expression *voisin* mesure la distance entre l'observation et une observation particulière pendant la période allant de COL&deb à COL&fin. Ainsi, si deb=3 et fin=50, alors le voisinage est calculé sur la période allant de COL2 à COL50.

```
%MACRO voisin (filin, filout,ident,valeur,deb,fin) ;

/* ANALYSE DU VOISINAGE

filin      : DATA SAS en entrée avec COL1, COL2, ...
filout     : DATA SAS en sortie.
ident      : identifiant alphanumérique des observations du tableau filin
valeur     : valeur de l'identifiant de la ligne objet de l'étude
              analogique
deb, fin   : le voisinage se calcule entre COL&deb et COL&fin */

* Ajout de la monographie à toutes les observations ;

DATA travail;
SET &filin ;
mono=0;
IF &ident="&valeur" THEN mono=1;
PROC SORT ;
BY DESCENDING mono;

DATA travail;
SET travail;
RETAIN mono&deb-mono&fin;
ARRAY are(*) mono&deb-mono&fin;
ARRAY vec(*) col&deb-col&fin;
IF _N_=1 THEN DO;
    DO i=1 TO DIM(are);
        are(i)=vec(i);
    END;
END ;
```

```

/* dis1 = distance de la monographie à chaque observation */

DATA travail ;
SET travail;
dis1 = 0;
somvec=SUM(OF col&deb-col&fin);
sommono=SUM(OF mono&deb-mono&fin);
ARRAY arvec(*) col&deb-col&fin;
ARRAY armono(*) mono&deb-mono&fin;
DO i=1 TO DIM(arvec);
dis1 = dis1+((arvec(i)/somvec)-(armono(i)/sommono))**2;
END;

PROC SORT;
BY dis1;

* Ajout des _N_-1 premiers voisins à la _N_ème observation */

DATA travail ;
SET travail ;
RETAIN cum&deb-cum&fin 0;
ARRAY are(*) col&deb-col&fin;
ARRAY arcum(*) cum&deb-cum&fin;
IF _N_ = 1 THEN DO ;
    DO i=1 TO DIM(arcum);
    arcum(i)=0;
    END;
END ;
IF _N_ NE 1 THEN DO ;
    DO i=1 TO DIM(arcum);
    arcum(i)=arcum(i)+are(i);
    END;
END ;

/* dis2=distance de la monographie aux (_N_-1)ers voisins */

DATA &filout ;
SET travail;
dis2=0;
IF _N_ NE 1 THEN DO ;
    ARRAY armono(*) mono&deb-mono&fin;
    ARRAY arcum(*) cum&deb-cum&fin;
    somcum=SUM (OF cum&deb-cum&fin);
    DO i=1 TO DIM(arcum);
    dis2=dis2+((arcum(i)/somcum)-(armono(i)/sommono))**2;
    END;
voisin=_N_-1;
END;

PROC SORT;
BY dis1;
PROC PRINT ;
VAR &ident dis1 dis2;
PROC PLOT;
PLOT dis1*voisin=&ident $ &ident;
PROC PLOT;
PLOT dis2*voisin=&ident $ &ident;

%MEND;

```