LES CAHIERS DE L'ANALYSE DES DONNÉES

J.-P. BENZÉCRI
F. BENZÉCRI

Sources de programmes d'analyse de données en langage PASCAL : (III) : élaboration de tableaux divers (IIIC) : corrélation et régression

Les cahiers de l'analyse des données, tome 22, n° 3 (1997), p. 281-292

<http://www.numdam.org/item?id=CAD_1997__22_3_281_0>

© Les cahiers de l'analyse des données, Dunod, 1997, tous droits réservés.

L'accès aux archives de la revue « Les cahiers de l'analyse des données » implique l'accord avec les conditions générales d'utilisation (http://www.numdam.org/conditions). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM
Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
http://www.numdam.org/

Les Cahiers de l'Analyse des Données
Vol. XXII - 1997 - n°3 - pp.281-292.

SOURCES DE PROGRAMMES D'ANALYSE DE DONNÉES EN LANGAGE PASCAL: (III) : ÉLABORATION DE TABLEAUX DIVERS (IIIC) : CORRÉLATION ET RÉGRESSION

[SOURCES PASCAL (IIIC)]

J.-P. & F. BENZÉCRI

1 Méthode et calculs

1.1 L'art de la régression

Dans les calculs usuels de corrélation et de régression, on considère un ensemble J de variables, définies sur un même ensemble I d'individus; I étant muni d'une loi de probabilité pour laquelle, ordinairement, les individus, i, ont tous la même masse: 1/cardI.

Si on en retranche sa moyenne, chaque variable est réduite à avoir moyenne nulle; puis, en divisant par l'écart type, on obtient une variable dite: centrée réduite; i.e., de moyenne nulle et de variance 1. Ainsi transformé, l'ensemble J des variables, devient un ensemble de vecteurs unitaires d'un même espace euclidien: l'espace des fonctions de moyenne nulle sur I; et les coefficients de corrélation s'interprètent comme des cosinus.

Dans ce cadre géométrique, la régression d'une variable j1 en fonction d'un ensemble Jx d'autres variables, dites explicatives, n'est autre que la projection du vecteur unitaire, associé à j1, sur le sous-espace engendré par les vecteurs associés aux variables jx de Jx: la régression est d'autant meilleure que le cosinus de l'angle formé par le vecteur j1 et sa projection est plus proche de 1.

Éventuellement, si, par simple combinaison linéaire des variables explicatives, l'approximation obtenue pour j1 n'est pas satisfaisante, on complète l'ensemble Jx, en y adjoignant des fonctions diverses des variables de base: monomes (carré, cube; ...); produits deux à deux; voire fonctions trigonométriques, facteurs issus d'une analyse de correspondance.

Il est clair que, d'une part, dans ce schéma géométrique, il n'y pas de limite claire à la complexité des formules d'approximation que l'on peut créer; et, d'autre part, il est difficile l'apprécier la stabilité de ces formules, donc leur validité; les modèles probabilistes, qu'on évoque d'ordinaire, étant mal assis.

Les cahiers de l'analyse des données 0339-3097/97/03/© Elsevier, Paris

Pour nous, l'analyse de correspondance, après codage si les données sont hétérogènes, offre, des liens entre variables, une vue plus sûre que celle issue des calculs de corrélation. Mais si, après examen des données, sur histogrammes, croisements et par l'analyse factorielle elle-même, on a acquis une vue claire de la forme que doit avoir la formule de régression simple adaptée au problème, il peut être loisible de chercher une telle formule. C'est suivant cet art de la régression qu'on a procédé dans l'article [CONSOMM. ÉLEC.].

1.2 Calculs de corrélation et de régression: le programme 'corel'

Dans sa version initiale (cf. [CORREL. JUXT.]) le programme "corel" se borne aux calculs de corrélation, effectués entre colonnes d'un tableau présenté sous l'un des formats acceptés par la procédure 'litab' (cf. IA§2). Pour un couple de variables: {j1, j2}, on obtient un coefficient de corrélation usuel; ainsi que deux formules de régression linéaire à une variable; respectivement: pour j1, en fonction de j2; et pour j2, en fonction de j1.

La seule originalité du calcul réside en ce que l'ensemble I, des individus, peut être restreint à un sous-ensemble Ir, en imposant à la valeur: k(i, j1), de la 1-ère variable, d'être comprise entre des bornes dont le choix est laissé à l'utilisateur. Ainsi qu'on l'explique dans [CORREL. JUXT.], l'on peut, notamment, en considérant plusieurs sous-ensembles {Ir, Ir', Ir",...}, afférents à des intervalles consécutifs de variation de j1, décrire, par une suite de droites de régression, une relation quasi fonctionnelle liant j2 à j1.

De plus, étant (comme l'analyse factorielle) fondés sur des propriétés d'inertie, les calculs de corrélation et d'ajustement linéaire tendent à être dominés par les éléments les plus écartés. En analyse des correspondances, l'influence de tels éléments, sur un facteur, peut et doit être contrôlée sur la colonne CTR. Au contraire, les calculs de régression sont souvent effectués sans contrôle, comme machinalement.

Or l'article cité propose un exemple simple où, la variable à expliquer étant liée strictement à la variable explicative par une relation linéaire par morceaux, le signe du coefficient de corrélation s'inverse selon qu'on considère le domaine des valeurs centrales; ou l'ensemble I tout entier; le calcul étant alors dominé par les éléments périphériques. D'où l'intérêt de considérer les données en restreignant diversement I.

Quant à la régression linéaire en fonction de plusieurs variables (ou à la régression polynomiale, en fonction d'une seule), nous en avons d'emblée montré les aléats. Cependant, l'étude de [CONSOMM. ÉLEC.] nous a montré qu'elle pouvait être intéressante et sure, pourvu qu'on examine minutieusement les données. Et c'est pourquoi nous avons étendu les fonctions du programme 'corel'.

De façon précise, la colonne: j1 étant prise comme variable à expliquer, l'on considère l'approximation linéaire de j1, en fonction de j2, comme n'étant qu'une première approximation, susceptible d'être complétée par des termes linéaires en d'autres variables; le nombre total des variables explicatives étant, arbitrairement limité à 5.

Pour la régularité de l'algorithme, il est commode de noter la y-ème formule de régression sous la forme:

k(i, j1) - m1 ≈ cov * apry(i) ;

Dans cette formule: m1 est la moyenne de: j1, sur l'ensemble restreint: Ir; apry, est une combinaison linéaire des y variables explicatives (de rang: 1, 2, ..., y} qui est assujétie à avoir moyenne nulle et variance 1; et est, de plus, choisie pour avoir la corrélation maxima avec la colonne j1; enfin, avec ces restrictions, le coefficient: cov, n'est autre que la covariance de la variable à expliquer et de apry.

En particulier: apr1 est la variable j2, diminuée de sa moyenne et divisée par son écart-type. Et les combinaisons successives: {apr2, apr3, ...} sont définies chacune comme combinaison linéaire de l'approximation linéaire précédente, apr(y-1), et de la variable: jy qu'on a choisi d'introduire; les coefficients étant calculés par les formules rappelées dans [CONSOMM. ÉLEC.].

Enfin, après le calcul d'une approximation: apry, le croisement de: apry (mise en abscisse) et de j1 (ordonnée) s'affiche à l'écran, comme un nuage de points; lequel, pour une régression parfaite se réduirait à une droite (ou, éventuellement, à un arc de courbe).

Au point où nous sommes parvenus, nous préférons proposer au lecteur le listage même de 'corel'; plutôt que d'introduire des notations mathématiques différentes de celles du langage: Pascal.

2 Listage du programme: 'corel'

Les notations étant celles posées dans IA, et développées dans les chapitres suivants, on se bornera à de brefs commentaires

2.1 Déclarations et procédures

La procédure: 'litab' (contenue dans l'unité séparée: {$Ulire}, cf. IA§2), constitue un premier segment: {$S entr}, qui n'est appelé que pour choisir, puis lire, le tableau des données.

Vient ensuite un segment: {$S partout}, dont le nom rappelle que les procédures en sont constamment appelées dans les calculs de régression et corrélation.

```
program corel;
uses memtypes,quickdraw,osintf,toolintf,sane,uver5;
var i,il,j,jp,js,j1,j2,carti,cartj,er1,ry,jy:integer;
ptot,pgen,pret,m1,m2,my,co1,co2,S1,S2,Sy,V1,V2,Vy,cov,res,cos,c2y,c1y,
ko2,koy,infl,supl,minl,maxl:extended;
sg1i,sg2j:pzgi;k1ji:kji;
nomba,nomf,titre,sig1,sig2,sigy:string;
repc,rpb,rpf,rpv,rpx,rpz,rpy,rpp:char;ft:text;
t1k:tk;s1i,s2j,paux:ptr;
{$S entr} {$U ulire5}
procedure litab(var pj:ptr;pk,pn,pt:ptr;var rc:char;
     var ic,jc:integer;var pi:ptr);external;
{$S partout}
procedure enumerer;begin
  writeln('ci-dessous sont rappelés les num et sigles des colonnes');
   for j:=1 to cartj do begin write(j:3,sigler(sg2j^[j]):5,';');
     if (j mod 9=0) then begin writeln;
        if ((j mod 207=0) and (j<cartj)) then begin
           write('pour afficher la suite de l''ensemble entrer *');
           readln(rpv) end end end ;
   if not (cartj mod 9=0) then writeln;end;
```

La procédure 'énumérer', qui affiche à l'écran la suite des colonnes, avec leurs numéros et sigles, sert à choisir les variables à traiter.

```
procedure masser;begin
  if (rpv='O') then for i:=1 to carti do k1ji[0]^[i]:=k1ji[jp]^[i]
                    else for i:=1 to carti do k1ji[0]^[i]:=1;end;
procedure toter;begin ptot:=0;
  for i:=1 to carti do ptot:= ptot+k1ji[0]^[i];end;
function moy(ja:integer):real;begin res:=0;
  for i:=1 to carti do res:=res+(k1ji[0]^[i]*k1ji[ja]^[i]);
  moy:=res/ptot;end;
function com(ja,jb:integer):real;begin res:=0;
  for i:=1 to carti do res:=res+(k1ji[0]^[i]*k1ji[ja]^[i]*k1ji[jb]^[i]);
  com:=res/ptot;end;
procedure signaler;begin sig2:=sigler(sg2j^[js]);
   writeln('variance de ',sig2,' = 0');
   if (rpx='O') then writeln(ft,'variance de ',sig2,' = 0');end;
```

La procédure: 'masser' met dans la colonne de rang zéro (colonne auxiliaire réservée par 'litab'; et utilisée, notamment, par: 'qori') la suite des poids des individus de I: selon le choix de l'utilisateur, les poids sont pris dans une colonne: jp, du tableau des données; ou tous fixés à un.

Ultérieurement, sont mis à zéro les poids des individus exclus du fait de bornes imposées à la variable j1. Et l'on calcule, par 'masser' la masse totale des individus retenus.

Les procédures: 'moy', et: 'com', calculent, respectivement, la moyenne de la colonne de rang: ja; et le comomoment des colonnes: {ja, jb}.

Éventuellement, par 'signaler', on avertit l'utilisateur qu'une variable: js, ayant moyenne nulle, est impropre aux calculs de régression; et, si une listage doit être créé (cas: rpx='O'), cette mention y est portée.

```
{$S dessiner}
procedure planer;
  var fond:rect;P0:point;Pch,Pcv:integer;maxa,mina,kf1,kfa,xh,yv:extended;
  begin
  mina:=k1ji[cartj+1]^[i1];maxa:=mina;
  for i:=il to carti do
     if not(k1ji[0]^[i]=0) then begin res:=k1ji[cartj+1]^[i];
     if (res<mina) then mina:=res;if (maxa<res) then maxa:=res;end;
  write('pour afficher le plan de croisement entrer * ');readln(rpp);
  setrect(fond,0,0,600,500);eraserect(fond);
  if (max1=min1) then kf1:=0 else kf1:=-245/(max1-min1);
  if (maxa=mina) then kfa:=0 else kfa:= 470/(maxa-mina);
  Pch:=2+round(-kfa*mina);Pcv:=10+round(-kf1*max1);
  if (min1<=0) and (0<=max1) and (mina<=0) and (0<=maxa)
     then begin moveto(Pch,Pcv);drawstring('+') end;
  if (0<=max1) and (min1<=0)
     then begin moveto(470,Pcv);drawstring(sig1) end;
  if (mina<=0) and (0<=maxa)
     then begin moveto(Pch,12);drawstring('apr');drawchar(chr(48+ry)) end;
  for i:=1 to carti do if not(k1ji[0]^[i]=0) then begin
    yv:=k1ji[j1]^[i];xh:=k1ji[cartj+1]^[i];
    Pch:=2+round(kfa*(xh-mina));
    Pcv:=10+round(kf1*(yv-max1));
    moveto(Pch,Pcv);drawchar('.') end;
  moveto(2,264);
  drawstring('abscisse: ');drwfl(mina);
  drawstring(concat(' ≤ apr',chr(48+ry),' ≤ '));drwfl(maxa);
  drawstring(' ; ordonnée: ');drwfl(min1);
  drawstring(concat(' ≤ ',sig1,' ≤ '));drwfl(max1);
  getpen(P0);moveto(2,P0.v+11);
  drawstring('pour quitter le plan croisant apr');drawchar(chr(48+ry));
  drawstring(concat(';horiz) et ',sig1,' (n° '));drwnum(j1);
  drawstring(';vertic) entrer * ');
  lir(rpp);eraserect(fond);moveto(5,10);writeln;
  end;
```

Le troisième segment: {$S dessiner}, est constitué de la procédure: 'planer'; laquelle n'est appelée que pour afficher le plan croisant la variable à expliquer avec l'une de ses approximations linéaires successives: apry, (cf. §1.2). Relativement à celles déjà considérées dans d'autres chapitres (cf. IC§1.2.4 et IIIA§4.2) 'planer' est très simple: elle n'affiche qu'un seul ensemble (I: réduit éventuellement à Ir par des bornes imposées à j1); et les éléments sont marqués, non par un sigle ou une lettre, mais par un point: '.' .

On a repris, avec des modifications minimes, la procédure 'planer' de 'zrang'. Dans la procédure 'planer' de 'corel', l'affichage, est inséré entre les instructions d'appel et d'effaçage du graphique. La combinaison: apr, considérée a été logée dans une colonne, de rang: cartj+1, au-delà du tableau des données. La colonne de rang zéro indique les éléments retenus (k1ji[0]^[i] ≠ 0). Les bornes: {min1, max1}, de la variable à expliquer (rang: j1, sigle: sig1), ont été calculées avant d'appeler 'planer'. Il reste à calculer les bornes: {mina, maxa} de la combinaison: apr. Pour les échelles horizontale et verticale, la place de l'origine, l'inscription de sigles à l'extrémité des axes, etc., on procède comme dans 'zrang'.

2.2 Le programme principal et corrélation et régression

Nous considérons d'abord, au §2.2.1, le programme tel qu'il était avant l'adjonction du calcul d'approximations de rang supérieur; ces approximations font l'objet du §2.2.2.

2.2.1 Calculs de corrélation entre deux variables de base

```
begin Benzecri;rpz:='O';
writeln('ce programme calcule des corrélations et régressions entre colonnes');
      while not(rpz='N') do begin
carti:=1;
litab(s2j,@t1k,@nomba,@titre,repc,carti,cartj,s1i);unloadseg(@litab);
rpz:=repc;
if (repc='O') then begin
  writeln(titre);
  for j:=0 to cartj do k1ji[j]:=pti(t1k[j]);
  paux:=newptr(4*(carti+1));k1ji[cartj+1]:=pti(paux);
  sg1i:=pzgi(s1i);sg2j:=pzgi(s2j);enumerer;
  write('y a-t-il une colonne de poids O ou N ');readln(rpv);
  if (rpv='O') then begin er1:=0;rpv:='N';
    while not ((er1=6) or (rpv='O')) do begin rpv:='O';
      write('numéro de la colonne de poids = ');readln(jp);
      if (jp<0) then jp:=1;if (cartj<jp) then jp:=cartj;ptot:=0;
      for i:=1 to carti do begin ptot:=ptot+k1ji[jp]^[i];
         if (k1ji[jp]^[i]<0) then rpv:='N' end;
      if (ptot=0) then rpv:='N';
      if (rpv='N') then begin er1:=er1+1;
        writeln('ERREUR la colonne a des valeurs < 0 ou est nulle') end
        else begin
        write('ce choix est il confirmé O ou N '); readln(rpv) end end end;
  masser;toter;writeln('ptot = ',ptot:8);pgen:=ptot;rpf:='O';
```

Le programme principal est compris dans une boucle générale:

while not(rpz='N'), qui commence, avec 'litab', par le choix et l'entrée du tableau de base à traiter.

Si le choix s'est fixé sur un tableau présent dans un disque accessible, (repc='O'), le traitement débute, comme à l'ordinaire, en donnant aux pointeurs le type qui permet de lire les nombres (colonnes: k1ji[j]); ou les sigles (sg1i, pour les lignes; sg2j, pour les colonnes).

Mais de plus, on sait qu'aux cartj colonnes de base, est ajoutée une colonne (de rang cartj+1) destinée à recevoir la combinaison: apry, servant à poursuivre la régression. Et, 'litab' ayant été appelée avec: carti:=1, est réservée la place d'une ligne de rang (carti+1) où seront marquées les variables pouvant servir à poursuivre la régression (cf. *infra*, §2.2.2).

Éventuellement (rpv='O'), l'utilisateur spécifie, dans un dialogue, une colonne: jp, de pondération; on vérifie qu'il n'y a pas de poids négatif et que le poids total: ptot, n'est pas nul. Ainsi qu'on l'a dit au §2.1, la procédure 'masser' range, dans la colonne zéro, les poids adoptés; dont le total est effectué par 'toter'.

```
write('faut il créer un listage des corrélations calculées O ou N ');
readln(rpx);if not(rpx='N') then rpx:='O';
if (rpx='O') then begin
     nomf:=concat(nomba,'relx');
     if (rpv='O') then begin
        sig1:=sigler(sg2j^[jp]);nomf:=concat(nomf,sig1);end;
     rewrite(ft,nomf);
     writeln(ft,titre);
     if not(rpv='O') then
        writeln(ft,'il n''y a pas de colonne de pondération');
     if (rpv='O') then
        writeln(ft,'colonne de pondération: col.n°',jp:3,' :',sig1);end;
```

Bien que les corrélations calculées s'affichent à l'écran, l'utilisateur demande, ordinairement, que soit créé un listage. Celui-ci est placé dans le même dossier que le tableau de base; le nom reçoit le suffixe:'relx'; auquel s'ajoute, s'il y a lieu, le sigle de la colonne de pondération. On évite ainsi que les calculs effectués lors d'un second appel, avec des pondérations différentes, n'effacent ceux déjà enregistrés.

```
while not (rpf='N') do begin enumerer;
   write('numéro de la colonne à expliquer = ');readln(j1);
   write('numéro de la colonne explicative = ');readln(j2);
   if (j1<1) then j1:=1;if (cartj<j1) then j1:=cartj;
   if (j2<1) then j2:=1;if (cartj<j2) then j2:=cartj;
    sig1:=sigler(sg2j^[j1]);sig2:=sigler(sg2j^[j2]);
   write('faut-il assujétir',sig1:5,' à des bornes O ou N ');
   readln(rpb);masser;toter;
   if (rpb='O') then begin
     write('borne inf de',sig1:5,' = ');readln(inf1);
     write('borne sup de',sig1:5,' = ');readln(sup1);
     if (sup1<inf1) then sup1:=inf1;
     for i:=1 to carti do
       if (klji[j1]^[i]<inf1) then klji[0]^[i]:=0 else
         if (klji[j1]^[i]>sup1) then klji[0]^[i]:=0;
     toter;writeln('ptot = ',ptot:8);
     if (ptot=0) then begin
       writeln('Les Bornes, étant Incompatibles, seront ignorées');
       rpb:='N';masser;toter end end;
```

Commence alors, au sein du traitement, d'un tableau de base donné, une boucle: while not (rpf='N') do begin..., où est traitée une variable: j1, d'abord associée à une seconde variable: j2; puis éventuellement expliquée par régression en fonction d'autres variables (cf. §2.2.2).

L'utilisateur peut spécifier des bornes pour j1; le programme vérifie que la masse totale restante n'est pas nulle; si tel est le cas, le traitement de la colonne j1 se poursuit sans tenir compte des bornes.

L'on calcule alors les moyennes: {m1, m2} des colonnes: {j1, j2}; ainsi que les variances et covariances. Puis, sous réserve que ne soit pas nul le produit, S1*S2, des écarts-type (les exceptions: S1=0 ou S2=0 étant dûment signalées), on calcule le coefficient de corrélation: cos, dont la valeur est ramenée dans l'intervalle (-1, +1); si l'imprécision des calculs l'en a fait sortir.

```
m1:=moy(j1);V1:=com(j1,j1)-sqr(m1)
if not(V1>0) then S1:=0 else S1:=sqrt(V1);
m2:=moy(j2);V2:=com(j2,j2)-sqr(m2);
if not(V2>0) then S2:=0 else S2:=sqrt(V2);
res:=com(j1,j2);cov:=res-m1*m2;cos:=S1*S2;
if not(cos=0) then cos:=cov/cos;pret:=ptot/pgen;
if (cos<-1) then cos:=-1;if (1<cos) then cos:=1;
writeln('corr(',sig1,',',sig2,') = ',cos:8,' ; poids retenu = ',pret:8);
if (rpx='O') then begin
   writeln(ft,'corrélation entre col',j1,': ',sig1,' et col',j2,': ',sig2);
   if (rpb='O') then begin
     writeln(ft,'bornes de ',sig1,' : inf = ',inf1:8,' ; sup = ',sup1:8);
      writeln(ft,'poids de l''intervalle retenu = ',pret:8) end;
   writeln(ft,'corr(',sig1,',',sig2,') = ',cos:8) end;
if not(cos=0) then begin col:=cov/V1;co2:=cov/V2;
 writeln(sig2,' - ',m2:8,' ≈ ',co1:8,' * (',sig1,' - ',m1:8,')');
 writeln(sig1,' - ',m1:8,' ≈ ',co2:8,' * (',sig2,' - ',m2:8,')');
 if (rpx='O') then begin
  writeln(ft,sig2,' - ',m2:8,' ≈ ',co1:8,' * (',sig1,' - ',m1:8,')');
  writeln(ft,sig1,' - ',m1:8,' ≈ ',co2:8,' * (',sig2,' - ',m2:8,')')
  end end;
if (S1=0) then begin js:=j1;signaler end;
if (S2=0) then begin js:=j2;signaler end;
```

Les formules de régression linéaires s'affichent à l'écran; et sont, s'il y a lieu, copiées sur le listage 'relx'.

2.2.2 Approximations de rang supérieur

```
ry:=1;rpy:='*';
if ((S1=0) or (S2=0) or (cos=1) or (cos=-1)) then rpy:='N';
if (rpy='*') then begin
   for j:=1 to cartj do k1ji[j]^[carti+1]:=1;
   k1ji[j1]^[carti+1]:=0;k1ji[j2]^[carti+1]:=0;
   k1ji[cartj+1]^[carti+1]:=cartj-2;end;
```

L'entrée dans les approximations de rang supérieur est régie par le caractère: rpy. Sont d'abord exclus divers cas:

S1=0, il n'y a rien à expliquer, la colonne j1 étant de variance nulle;

S2=0, la variable j2 n'ayant rien apporté, la régression de j1 est à reprendre à son début;

cos=1, ou -1, une régression parfaite de j1 a déjà té obtenue avce j2.

Après quoi, l'on note par 1, dans la ligne: cari+1 (cf. *supra*), les variables pouvant contribuer à améliorer la régression; j1 et j2, étant exclues (valeur 0), le choix reste ouvert entre cartj-2 variables; le nombre des possibles étant désormais noté dans la case: k1ji[cartj+1]^[carti+1].

Commence alors une boucle:

```
while not((ry=5) or (rpy='N') or (k1ji[cartj+1]^[carti+1]=0)) do begin
```

qui exécute les approximations successives, de rang: 2, 3, ...5; sous réserve que l'on n'ait pas déjà atteint le rang 5, qu'il reste des variables disponibles; et que l'utilisateur n'ait pas formulé son refus, par: rpy='N'.

```
while not((ry=5) or (rpy='N') or (k1ji[cartj+1]^[carti+1]=0))
  do begin ry:=ry+1;
  write('faut-il introduire une ',chr(48+ry),'-ème col explicative pour: ',sig1,' O ou N '
  readln(rpy);
  if not(rpy='N') then begin rpy:='*';er1:=0;
    if (ry=2) then begin res:=1/S2;
       for i:=1 to carti do k1ji[cartj+1]^[i]:=res*(k1ji[j2]^[i]-m2);
       writeln('apr1 = ',res:8,' * (',sig2,' - ',m2:8,')');end;
    while not((rpy='O') or (er1=6)) do begin enumerer;
      write('numéro de la nouvelle col explicative = ');readln(jy);
      if (jy<1) then jy:=1;if (cartj<jy) then jy:=cartj;
      if (k1ji[jy]^[carti+1]=0) then begin jy:=1;
        while not((k1ji[jy]^[carti+1]=1) or (jy=cartj)) do jy:=jy+1;end;
      sigy:=sigler(sg2j^[jy]);
      write('le choix de la colonne: ',sigy,' est-il confirmé O ou N ');
      readln(rpy);
      if (rpy='O') then begin
        k1ji[jy]^[carti+1]:=0;
        k1ji[cartj+1]^[carti+1]:=k1ji[cartj+1]^[carti+1]-1;
        my:=moy(jy);Vy:=com(jy,jy)-sqr(my);
        if not(Vy>0) then Sy:=0 else Sy:=sqrt(Vy);
        c1y:=Sy;
        if not(Sy=0) then begin
          res:=com(j1,jy);res:=res-m1*my;c1y:=res/(S1*Sy);
          if (c1y<-1) then c1y:=-1;if (1<c1y) then c1y:=1;
          writeln('corr(',sig1,',',sigy,') = ',c1y:8);
          res:=com(jy,cartj+1);c2y:=res/Sy;
          if (c2y<-1) then c2y:=-1;if (1<c2y) then c2y:=1;
          if ((c2y=1) or (c2y=-1)) then begin er1:=er1+1;rpy:='N';
            writeln('ERREUR: la col: ',sigy,'est liée à apr',chr(47+ry),'cor=',c2y:8);
        end;end;
        if (c1y=0) then begin er1:=er1+1;rpy:='N';
          writeln('ERREUR: la col: ',sigy,'n''est pas corrélée à: ',sig1);end;
  end;end;end;
```

Si (ry=2), il convient de réduire la variable: j2, déjà utilisée; et de la ranger dans la colonne: cartj+1. On entre alors dans le choix d'une nouvelle variable explicative: jy. Si la proposition de l'utilisateur est inacceptable, le programme la corrige; puis (la correction étant acceptée s'il y a lieu), il faut considérer, de plus près, si la colonne: jy, n'est pas elle-même inadéquate: parce qu'elle a variance nulle; ou est parfaitement corrélée à l'approximation déjà obtenue (à laquelle on peut donc la substituer, sans autre calcul); ou encore n'est aucunement corrélée à j1. Dans tous les cas (que jy soit acceptée ou rejetée), elle ne sera plus disponible pour des régressions ultérieures; d'où l'instruction préalable: k1ji[jy]^[carti+1]:=0.

En cas de succès, une instruction composée: if (rpy='O') then begin..., commande l'introduction de la colonne explicative: jy.

D'abord, si ry=2, on doit déterminer les bornes de l'intervalle de variation de la variable j2 introduite comme première variable explicative. Et on affiche, par 'planer', le croisement de j1 avec j2 (plus exactement, avec: apr1, qui n'est autre que: j2 centrée et normée). Puis le listage 'relx' est complété.

```
if (rpy='O') then begin
  if (ry=2) then begin i1:=0;
     for i:=1 to carti do begin
       if not(k1ji[0]^[i]=0) then begin res:=k1ji[j1]^[i];
          if (i1=0) then begin i1:=i;min1:=res;max1:=res end;
          if (res<min1) then min1:=res;if (max1<res) then max1:=res;end;end;
     ry:=1;planer;unloadseg(@planer);ry:=2;end;
  if (rpx='O') then begin
     if (ry=2) then begin res:=1/S2;
        writeln(ft,'apr1 = ',res:8,' * (',sig2,' - ',m2:8,')');end;
     writeln(ft,chr(48+ry),'-ème col explicative: col',jy,': ',sigy);
     writeln(ft,'corr(',sig1,',',sigy,') = ',c1y:8) end;
```

La suite du calcul consiste essentiellement dans l'application des formules du §2.3 de l'article [CONSOMM. ÉLEC.]. De façon précise, On a les trois coefficients de corrélation:

c1y = corr(sigy,sig1) ; c2y = corr(sigy,apr(ry-1));cos: corr(sig1,apr(ry-1));

Avec les notations de l'article, c2y n'est autre que le cosinus (C) de l'angle formé par les axes des deux variables explicatives, plus exactement de l'approxiamtion précédente: apr(ry-1), et de la nouvelle variable: jy.

D'où, selon les formules, le calcul des coeffiicients de régression:

koy:=((1+(C*C/(S*S)))*c1y)-((C/(S*S))*cos;

ko2:=((1+(C*C/(S*S)))*cos)-((C/(S*S))*c1y;

ko2, pour apr(ry-1); et koy, pour jy; ce dernier devant être corrigé parce que: jy, n'est pas normalisée.

La nouvelle approximation: apr(ry), est ensuite normalisée. Et, après affichage (et copie sur: relx) des coefficients trouvés, le traitement de jy s'achève en affichant le croisement de: apr(ry), avec: j1.

```
  res:=c2y/(1-(c2y*c2y));
  koy:=((1+(c2y*res))*c1y)-(res*cos);
  ko2:=((1+(c2y*res))*cos)-(res*c1y);res:=koy/Sy;
  for i:=1 to carti do
     k1ji[cartj+1]^[i]:=(res*(k1ji[jy]^[i]-my))+(ko2*k1ji[cartj+1]^[i]);
  V2:=com(cartj+1,cartj+1);if (V2<0) then V2:=0;
  S2:=sqrt(V2);if (S2=0) then S2:=1;res:=1/S2;
  for i:=1 to carti do k1ji[cartj+1]^[i]:=res*k1ji[cartj+1]^[i];
  ko2:=ko2/S2;koy:=koy/(Sy*S2);
  writeln('apr',chr(48+ry),' = ',ko2:8,' * apr',chr(47+ry),' + ',koy:8,' * (',sigy,' - ',my:8,')');
  cov:=com(j1,cartj+1);cos:=cov/S1;
  if (cos<-1) then cos:=-1;if (1<cos) then cos:=1;
  writeln(sig1,' - ',m1:8,' ≈ ',cov:8,' * apr',chr(48+ry));
  writeln('corr(',sig1,',apr',chr(48+ry),') = ',cos:8);
  if (rpx='O') then begin
    writeln(ft,'apr',chr(48+ry),' = ',ko2:8,' * apr',chr(47+ry),' + ',koy:8,' * (',sigy,' - ',my:8,')');
    writeln(ft,sig1,' - ',m1:8,' ≈ ',cov:8,' * apr',chr(48+ry));
    writeln(ft,'corr(',sig1,',apr',chr(48+ry),') = ',cos:8);end;
  planer;unloadseg(@planer);
end;end;
```

```
    writeln(ft);
    write('faut il expliquer un autre colonne O ou N ');readln(rpf) end;
  if (rpx='O') then close(ft);
  for j:=0 to cartj do dispose(tlk[j]);
  dispose(s1i);dispose(s2j);dispose(paux);
  write('faut il calculer des corr sur un autre tableau O ou N ');
  readln(rpz) end;    end  ;
readln(rpf);end.
```

Au sortir de la boucle des approximations successives, le traitement de la variable à expliquer: j1, est lui-même achevé. Éventuellement, on recommence avec une autre colonne (voire, avec la même colonne; mais assujétie à d'autres bornes). Sinon la mémoire est libérée; et on propose à l'utilisateur de considérer un autre tableau: ce qui implique de renter dans la boucle générale de 'corel'.

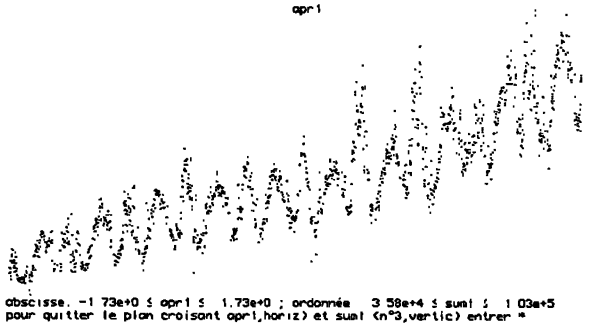
3 Exemple: l'estimation de la consommation d'électricité

Comme exemple, on a repris les données de [CORREL. JUXT.]. On a procédé de deux manières: d'une part, en introduisant d'abord la date: num; puis la température (représentée ici par le facteur: axe1, issu d'une analyse après codage barycentrique); d'autre part, en procédant dans l'ordre inverse. Les résultats s'accordent pleinement entre eux et avec ceux de l'article.

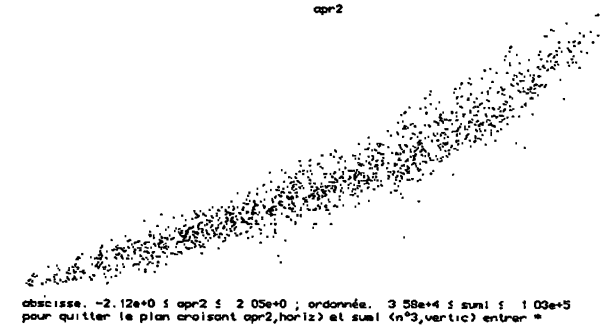
```
électricité en climat extrême
il n'y a pas de colonne de pondération
corrélation entre col       3: suml et col       1: num
corr(suml,num) =  8.7724851e-1
num -  1.8107802e+3 ≈  6.6609779e-2 * (suml -  6.2387848e+4)
suml -  6.2387848e+4 ≈  1.1553333e+1 * (num -  1.8107802e+3)
apr1 =  9.5673052e-4 * (num -  1.8107802e+3)
2-ème col explicative: col       4: axe1
corr(suml,axe1) =  4.2289210e-1
apr2 =  8.9804815e-1 * apr1 +  4.5219350e-1 * (axe1 - -3.9503899e-5)
suml -  6.2387848e+4 ≈  1.3174597e+4 * apr2
corr(suml,apr2) =  9.5706693e-1

corrélation entre col       3: suml et col       4: axe1
corr(suml,axe1) =  4.2289210e-1
axe1 - -3.9503899e-5 ≈  2.7190905e-5 * (suml -  6.2387848e+4)
suml -  6.2387848e+4 ≈  6.5771157e+3 * (axe1 - -3.9503899e-5)
apr1 =  1.1298243e+0 * (axe1 - -3.9503899e-5)
2-ème col explicative: col       1: num
corr(suml,num) =  8.7724851e-1
apr2 =  4.0023348e-1 * apr1 +  8.5919007e-4 * (num -  1.8107802e+3)
suml -  6.2387848e+4 ≈  1.3174597e+4 * apr2
corr(suml,apr2) =  9.5706693e-1
```

Outre le listage 'relx', ci-dessus, nous publions, tels qu'ils sont apparus à l'écran, d'une part, le croisement de la consommation: suml, avec la date: num; celle-ci étant convertie en une variable centrée normée: apr1, ou première approximation; et, d'autre part, le croisement avec l'approximation: apr2, à laquelle on s'est arrêté.

Dans le premier croisement, le lecteur de [CORREL. JUXT.] reconnaîtra,



pour chacune des dix années étudiées, les deux maxima liés, respectivement, au froid de l'Hiver et à la chaleur de l'Été.



Références bibliographiques

- J.-P. & F. BENZÉCRI : "Calculs de corrélation entre variables et juxtaposition de tableaux"; [CORREL. JUXT.]; in *CAD*, Vol.XIV, n°3, pp. 347-354; (1989);

J. de TIBEIRO : "Consomation d'électricité sous un climat extrême: estimation en fonction de la date et de la température", [CONSOMM. ÉLEC.]; in *CAD*, Vol.XXII, n°2, pp. 199-210; (1997).