LES CAHIERS DE L'ANALYSE DES DONNÉES

J.-P. BENZÉCRI
F. BENZÉCRI

Sources de programmes d'analyse de données en langage PASCAL : (I) : analyse des correspondances (IC) : tracé des graphiques plans

Les cahiers de l'analyse des données, tome 22, n° 1 (1997), p. 55-70

<http://www.numdam.org/item?id=CAD_1997__22_1_55_0>

Les Cahiers de l'Analyse des Données
Vol. XXII - 1997 n°1 pp.55 70.

SOURCES DE PROGRAMMES D'ANALYSE DE DONNÉES EN LANGAGE PASCAL: (I) : ANALYSE DES CORRESPONDANCES (IC) : TRACÉ DES GRAPHIQUES PLANS

[SOURCES PASCAL (IC)]

J.-P. & F. BENZÉCRI

0 Introduction: affichage à l'écran et composition en format de texte

Pour le tracé des graphiques plans croisant deux facteurs, nous avons deux programmes: d'une part, 'PlanF', qui affiche des graphiques à l'écran; d'autre part, 'PlanX', qui crée un listage contenant, en format de texte, les graphiques plans demandés par l'utilisateur.

Présentement, nous utilisons quasi exclusivement 'PlanF'.

Par 'PlanF', les éléments des ensembles à représenter s'affichent avec les coordonnées demandées, à un pixel près; tandis que 'PlanX' a pour unité de distance verticale, l'interligne; et pour unité horizontale, la largeur de l'espace attribué à un caractère.

De plus, les graphiques affichés par 'PlanF', une fois saisis comme des fichiers graphiques, peuvent être modifiés: en déplaçant des sigles superposés, couvrant de hachures une zone dense; encadrant des titres ou commentaires...

Et, tandis qu'un listage créé par 'PlanX' ne doit être affiché ou imprimé qu'avec une police de caractères ayant tous la même largeur, 'PlanF' peut attribuer, à chacun des ensembles à représenter, quelque police que ce soit, choisie par l'utilisateur dans la taille qu'il veut.

Mais, d'autre part, 'PlanX', ne superpose jamais de sigles: en créant le listage, il vérifie que chaque sigle qu'il y insère se loge dans un espace blanc assez large; et, si cela n'est pas, mentionne explicitement cette superposition virtuelle. De plus, 'PlanX' peut être aisément adapté à tout ordinateur; tandis que 'PlanF' utilise des ordres graphiques, qui ne sont pas codifiés dans un langage tel que PASCAL, mais propres à un système particulier.

Nous publirons donc, simultanément, les sources pour 'PlanF' et pour 'PlanX'

Les cahiers de l'analyse des données 0339 3097/97/01/$ 7 00/© Gauthier Villars

1 Création de graphiques dans une fenêtre de l'écran: le programme 'PlanF'

Le programme 'PlanF' se compose d'un programme principal, 'planF.p' et d'une unité séparée, 'uplanF.p'. Nous considérerons succesivement ces deux parties

1.1 Le programme principal: 'planF.p'

```
program planF;
uses memtypes,quickdraw,osintf,toolintf,sane,uver;
var nombas,nomf:str255;ch1,rec:char;er1,c:integer;
{$S segplan} {$U uplanF}
procedure plan(nomen:str255);external;
begin;rec:='O';benzecri;
while not (rec='N') do begin rec:='N';er1:=0;
  while not ((rec='O') or (er1=6)) do begin rec:='O';
    write('le fichier de base est ');
    readln(nombas);
    if not(setvol(@nombas,10)=noerr) then begin rec:='N';er1:=er1+1;
      writeln('ERREUR le disque manque ') end;
    if (er1=6) then writeln('après 6 erreurs on abandonne');
    if (rec='O') then begin
      for c:=1 to 2 do begin ch1:=chr(104+c);
        nomf:=concat(nombas,ch1,'Fac.w');
        if not(verif(@nomf)=2) then
           writeln('NB il n''y a pas de fichier des Facteurs pour',ch1:2) end;
      write('le nom du fichier de base est-il confirmé oui(O) ou non(N) ');
      readln(rec) end end;
  if (rec='O') then begin
    plan(nombas);unloadseg(@plan);
    write('faut-il poursuivre avec un autre fichier de base O ou N ');
    readln(rec) end;
end;
readln(rec);end.
```

Le programme principal consiste en une boucle: while not (rec='N')..., qui tourne aussi longtemps que l'utilisateur est à même de désigner un disque existant et ne refuse pas de poursuivre. De façon précise, il y a dans la boucle générale (externe), une seconde boucle (interne): while not (rec='O')...; puis l'instruction conditionnelle: if (rec='O') then..., où est appelée la procédure 'plan' (de l'unité 'uplan'), qui affiche des graphiques suivant les ordres de l'utilisateur.

Dans la boucle interne l'utilisateur est averti de l'absence éventuelle, de fichier de facteur pour un ensemble principal, Ip ou Jp (fichier avec suffixe: iFac.w, ou: jFac.w, cf. IB§2.1.3). Il se peut que, l'utilisateur ne voulant représenter que des ensembles supplémentaires (e.g. des ensembles de lignes ou colonnes supplémentaires adjointes au tableau de base), les fichiers {iFac, jFac} ne soient pas dans le dossier donné par l'utilisateur: mais celui-ci est du moins averti s'il a tapé un nom erroné.

Dans l'instruction: if (rec='O')..., après les tracés de 'plan', l'utilisateur peut demander à traiter un autre fichier de base; auquel cas la boucle externe est reprise; sinon, 'PlanF' se termine par la lecture d'un caractère.

1.2 Tracé des graphiques plans: la procédure 'plan'

```
UNIT uplanF5;
INTERFACE uses memtypes,quickdraw,osintf,toolintf,sane,uver5;
procedure plan(nomen:str255);
```

L'unité 'uplanF.p', qui ne comporte ni constante ni type qui lui soit propre (i.e. non déjà déclaré dans 'uver'; cf. IA§1), consiste exclusivement en la procédure de tracé 'plan'. Celle-ci, est appelée avec un seul argument: la chaîne de caractères 'nomen' qui donne le nom du tableau de base chosisi dans le programme principal 'PlanF'.

Après les déclarations des variables, la procédure 'plan' comprend trois parties, d'inégale complexité:

choix des ensembles susceptibles de figurer sur les graphiques (nous dirons, en bref: choix des ensembles possibles) et du mode de réprésentation adopté pour les éléments de chacun d'eux (sigles complets ou caractère unique etc...);

calcul des bornes des facteurs sur chacun des ensembles;

tracé des graphiques.

Cette dernière partie peut être schématisée par une double boucle: dans la boucle externe, l'utilisateur spécifie, parmi les ensembles possibles, ceux qu'il veut voir représenter dans une série de graphiques plans. À chaque itération de la boucle interne, un graphique de la série est défini par dialogue (choix des axes horizontal et vertical; échelle...); puis affiché à l'écran.

1.2.1 Déclarations de la procédure 'plan'

```
implementation procedure plan;
const afmax=10;emax=20;
var  riri:ricla;firi:file of integer;
  ch1,rpp,rpe,che,rpc,rpl,rph,rsc,rpf,chh,chv:char;nomf,mot:string;ptrcl:ptrc;
  rec0,recd:rect;
  a,ah,av,sh,sv,amax,fmax,u,i,c,cari,e,care,carde,Pch,
  Pcv,dir,er1,ver,nh,hi,rg:integer;
  res,co1,co2:real;tic:longint;
  ens:array[1..emax] of zigle;
  sge,sye:array[1..emax]of char;rng:array[1..emax] of integer;
  pre,fte,sze:array[1..emax] of integer;
  inf,sup:array[1..emax,1..afmax] of single;
  gch,hau,drt,bas,sss:single;
```

Quel que soit le nombre de facteurs disponibles dans les fichiers 'Fac.w', la procédure n'en considère pas plus que 'afmax' pour les graphiques. Le nombre des ensembles possibles est limité à 'emax' (parmi lesquels on peut avoir: 'i'; 'is'; 'j'; l'ensemble 'iq' des centres de classes d'une CAH, cf. IIC§3; etc.). Les noms de ces ensembles sont dans le tableau 'ens', de zigles; les tableaux {sge, sye, fte, sze} spécifient comment représenter ens[e]; pre[e]=1 si ens[e] est à représenter dans la série en cours. Les tableaux 'inf' et 'sup' donnent les bornes des facteurs sur chaque ens[e]. Le type 'ricla' sert à interpréter les données (sigle et facteurs) lues d'abord, pour chaque élément, dans 'firi', comme une suite d'entiers (cf. IB§2.1.4: procédure 'riclage'); etc.

1.2.2 Choix des ensembles susceptibles de figurer sur les graphiques

```
begin setrect(rec0,0,0,510,310);er1:=0;
rpe:='O';rpp:='O';rpc:='N';fmax:=afmax;ptrc1:=ptrc(@riri);
while not ((rpc='O') or (er1=6)) do begin
  for e:=1 to emax do begin fte[e]:=4;sze[e]:=9;end;
  eraserect(rec0);moveto(20,20);
  write('le nombre (1 a',emax:3,') des ensembles considérés est ');
  readln(care);if (care<1) then care:=1;if (emax<care) then care:=emax;
  e:=0;rpc:='O';
  while (e<care) and (rpc='O') do begin e:=e+1;
    write('le sigle de l''ensemble no ',e:2,' est ');
    readln(mot);if (4<length(mot)) then mot:=copy(mot,1,4);ens[e]:=zigler(mot);
    nomf:=concat(nomen,mot,'Fac.w');ver:=verif(@nomf);
    if (ver=2) then begin sge[e]:='N';
     write('faut-il prendre une fonte particulière pour ',sigler(ens[e]),' O ou N ');
     readln(rpf);if (rpf='O') then begin rpf:='N';
        while not (rpf='O') do begin
          write('nom de la fonte choisie = ');readln(mot);
          GetFnum(mot,fte[e]);
          writeln('le numéro de la fonte choisie est ',fte[e]);
          write('taille de la fonte choisie = ');readln(sze[e]);
          if (sze[e]<9) then sze[e]:=9;if (18<sze[e]) then sze[e]:=18;
          textfont(fte[e]);textsize(sze[e]);writeln;
          writeln('     ABCDE abcde 1234     ');readln(rpf);writeln;
          textfont(4);textsize(9);
          write('le choix de la fonte est-il confirmé (O) ou (N) ');
          readln(rpf);end;end;
     writeln('faut-il figurer les élém ',sigler(ens[e]),' par leur sigle(S)');
     writeln('par une seule lettre de leur sigle');
     writeln('choisie de la première(1), à la dernière(4)');
     write('ou par un caractère unique pour l''ensemble(U) ');readln(rp1);
     if not(rp1 in ['1','2','3','4','S','U']) then rp1:='S';
     if (rp1 in ['1','2','3','4']) then begin
         sge[e]:=rp1;rng[e]:=ord(rp1)-48 end;
     if (rp1='U') then begin sge[e]:='O';
         write('le caractère choisi pour ',sigler(ens[e]),' est ');
          readln(sye[e]) end end;
    if not(ver=2) then begin er1:=er1+1;
      writeln('ERREUR le fichier des facteurs manque pour ',sigler(ens[e]));
      if (er1=6) then begin
          writeln('après 6 erreurs on abandonne');
          tic:=TickCount;
          while (TickCount<tic+160) do res:=sqrt(TickCount) end
       else begin
          write('pour reprendre après cette erreur entrer une lettre ');
          readln(rpc) end;
       rpc:='N' end end;
  if (rpc='O') then begin
    write('ce choix est il confirmé oui(O) ou non(N) ');
    readln(rpc) end end;
```

Ce choix est compris dans la boucle: while not ((rpc='O') or (er1=6)) do. D'abord, par les instructions: fte[e]:=4;sze[e]:=9, la fonte n°4 (appelée 'Monaco' par le système) est attribuée à tous les ensembles, avec la taille 9. Et l'utilisateur fixe le nombre, 'care' des ensembles possibles. Commence alors une boucle: while (e<care) and (rpc='O') do...; laquelle, si aucun fichier de facteurs ne manque, tourne 'care' fois, jusqu'à la confirmation finale; avec, pour chaque ensemble, l'instruction composée: if (ver=2) then begin...

Mais si le fichier manque, l'instruction: if not(ver=2) then begin..., avertit l'utilisateur de son 'ERREUR'; et, si le nombre, er1, des erreurs faites est <6, renvoie au début de la boucle externe; while not ((rpc='O') or (er1=6)) do...; mais, pour er1=6, termine l'exécution de la procédure 'plan', en laissant affichée à l'écran la mention: 'après 6 erreurs on abandonne', aussi longtemps que tourne une boucle d'extraction de racine carrée, réglée sur l'horloge.

Reste donc à considérer, dans l'instruction composée: if (ver=2) then... , la spécification des formats demandés pour l'affichage d'un ensemble.

D'abord, si est demandée une fonte particulière (i.e. autre que 'Monaco' en taille 9), celle-ci est choisie dans la boucle: while not (rpf='O') do..., jusqu'à confirmation par rpf='O'. Il faut savoir que, dans le Macintosh, chaque fonte a, à la fois, un nom et un numéro; mais, s'il s'agit d'une fonte créée par l'utilisateur lui-même pour une langue requerrant des signes particuliers, etc., il vaut mieux la désigner par le nom (bien fixé) que par le numéro, qui, selon notre expérience, est attribué de façon inconstante selon le contenu du système. On notera qu'un échantillon de la fonte demandée est donné par l'instruction: writeln(' ABCDE abcde 1234 '). Après quoi, le dialogue se poursuit, avec la fonte usuelle (Monaco 9).

Ensuite, il faut choisir comment figurer les éléments. À cette question, on répondra, communément 'S', par leur sigle; mais s'il s'agit d'un ensemble dense d'individus, dont les sigles (e.g.: numéros de dossiers) n'évoquent rien, il suffit de marquer la distribution du nuage par un caractère unique 'U', généralement un point; mais parfois, afin de distinguer plusieurs ensembles, une barre convenablement orientées: {'|', '/', '\', '–'}. En effet, le programme 'soustab' permet de créer des fichiers de facteurs pour des sous-ensembles de I, J (ou autre) spécifiés par dialogue à l'écran, ou par fichier de commande.

Parfois, le sigle d'un élément, sans être totalement pourvu de sens, comporte, outre des chiffres arbitraires, un caractère particulier (lettre, chiffre ou autre) qui délimite divers sous-ensembles: e.g., la première lettre donne la diagnostic; la seconde, le lieu d'origine. C'est pourquoi est proposé le choix d'un caractère unique, défini par son rang dans le sigle.

Rien n'interdit à l'utilisateur de demander plusieurs fois le même ensemble, avec des formats d'affichage différents. Ainsi, en combinant des portions périphériques d'un graphique où sont écrits les sigles entiers, avec le centre d'un autre, où les éléments ne sont marqués que par un caratère, on obtient un plan comportant le maximum d'information lisible.

Le lecteur remarquera que la réponse 'S' (par le sigle) est choisie par défaut; que le rang, rng[e], du caractère à conserver, est calculé, suivant le code ASCII, comme la différence: ord(rpl)-48; etc.

1.2.3 Calcul des bornes des facteurs sur chacun des ensembles

```
if (rpc='O') then begin
 drawstring('afin de préparer le tracé on calcule les bornes des Facteurs');
 with riri do begin for e:=1 to care do begin
    for a:=1 to afmax do begin
       inf[e,a]:=100000;sup[e,a]:=-100000 end;
    nomf:=concat(nomen,sigler(ens[e]),'Fac.w');reset(firi,nomf);
    for u:=1 to titab-1 do read(firi,c);
    read(firi,cari);read(firi,c);amax:=c-2;dir:=7+(2*amax);
    if (amax<fmax) then fmax:=amax;
    for u:=1 to 3*(amax+2) do read(firi,c);
    for i:=1 to cari do begin
     for u:=1 to dir do read(firi,ptrcl^[u]);
     for a:=1 to fmax do begin
       if (Fac[a]<inf[e,a]) then inf[e,a]:=Fac[a];
       if (Fac[a]>sup[e,a]) then sup[e,a]:=Fac[a] end end;close(firi) end end;
```

Ainsi qu'on l'a annoncé au §1.2.1, deux tableaux, 'inf' et 'sup', reçoivent, respectivement, les minima et Maxima des facteurs sur chacun des ensembles possibles. Le nombre de ceux-ci étant 'care', le calcul des extrema consiste en une boucle: for e:=1 to care do...; chaque fichier: ens[e]Fac.w, doit être ouvert. On doit, d'abord, prendre garde à l'en-tête du fichier Fac.w, décrit dans IB§2.1.4. Les informations afférentes aux individus, lues, pour chacun de ceux-ci, comme une suite d'entiers (dont le nombre est: dir), sont comprises grâce à la structure 'ricla' (d'où l'en-tête: with riri do...).

Initialement, les extrema sont mis à des valeurs, ±100000, qui sortent nettement des limites usuelles de variation des facteurs (dont, en analyse des correspondances, la moyenne est 0 et la variance ≤1). Le nombre de facteurs considérés ne peut dépasser la constante afmax=10, déclarée en tête de 'plan.p'. Mais, de plus, on doit déterminer le nombre, 'fmax', des facteurs enregistrés pour tous les ensembles, et donc disponibles pour tout graphique demandé par l'utilisateur (cf. *infra*, §1.2.4.2).

1.2.4 Tracé des graphiques

Les tracés sont demandés par séries, la boucle: while not(rpe='N') do..., comprend la spécification d'une série, suivie d'une boucle de demandes de plans: while not(rpp='N') do...; chaque demande étant suivie par le tracé.

1.2.4.1 Choix des ensembles à représenter dans une série

```
while not(rpe='N') do begin eraserect(rec0);moveto(20,20);carde:=0;
 writeln('choix des ensembles à représenter parmi ceux considérés');
 for e:=1 to care do begin pre[e]:=0;
    write('faut il représenter ',sigler(ens[e]),' oui(O) ou non(N) ');
    readln(che);if not (che='N') then pre[e]:=1;carde:=carde+pre[e] end;
 if (carde=0) then pre[1]:=1;   rpp:='O';
```

On notera que si l'utilisateur n'a demandé aucun des ensembles possibles, la procédure impose que soit pris le premier de ceux-ci: pre[e]:=1.

1.2.4.2 Spécification d'un graphique au sein d'une série

```
while not(rpp='N') do begin eraserect(rec0);moveto(20,20);
 writeln('NB afin que s''affiche chaque ensemble entrer une lettre');
 writeln('de même, on met fin à l''affichage en entrant une lettre');
 writeln('s''il y a deux écrans, pour passer à l''écran inférieur, entrer une lettre');
 writeln('choix des axes du plan');
 writeln('NB: pour changer l''orientation d''un axe, donner son rang avec le signe -');
 write('le numéro de l''axe horizontal est ah = ');
 readln(ah);sh:=1;chh:='+';if (ah<0) then begin sh:=-1;ah:=-ah;chh:='-' end;
 if (ah<1) then ah:=1;if (fmax<ah) then ah:=fmax;
 write('le numéro de l''axe vertical est av = ');
 readln(av);sv:=1;chv:='+';if (av<0) then begin sv:=-1;av:=-av;chv:='-' end;
 if (av<1) then av:=1;if (fmax<av) then av:=fmax;
 write('faut-il adopter la même échelle sur les deux axes O ou N ');readln(rsc);
 write('faut-il afficher le plan sur deux écrans superposés O ou N ');
 readln(rph);nh:=1;if (rph='O') then nh:=2;
```

L'utilisateur choisit d'abord les facteurs afférents à chacun des deux axes. L'éventualité d'un changement de signe sert pour comparer les résultats de deux analyses, portant sur des données analogues, mais avec, e.g., un choix différent des éléments supplémentaires; et ayant produit certains facteurs similaires, au signe près; lequel, on le sait, est fixé aléatoirement par l'algorithme de diagonalisation.

Selon ce qui est demandé, le programme étale les nuages de points sur tout l'espace disponible; ou restreint l'une ou l'autre des deux dimensions (verticale ou horizontale) afin que l'échelle soit la même sur les deux axes. Le graphique peut être affiché sur un seul écran, comme un rectangle horizontal; ou sur deux écrans successifs; dont les images saisies sont assemblées, e.g., par 'MacPaint', pour obtenir un rectangle vartical .

1.2.4.3 Calcul des coefficients d'échelle pour le tracé

```
eraserect(rec0);
gch:=100000;hau:=100000;drt:=-100000;bas:=-100000;
for e:=1 to care do if (pre[e]=1) then begin
 if (inf[e,ah]<gch) then gch:=inf[e,ah];
 if (sup[e,ah]>drt) then drt:=sup[e,ah];
 if (-sup[e,av]<hau) then hau:=-sup[e,av];
 if (-inf[e,av]>bas) then bas:=-inf[e,av] end;
if (sh=-1) then begin sss:=drt;drt:=-gch;gch:=-sss end;
if (sv=-1) then begin sss:=hau;hau:=-bas;bas:=-sss end;
if (drt=gch) then co1:=0 else co1:=470/(drt-gch);
if (bas=hau) then co2:=0 else co2:=-nh*270/(bas-hau);
if ((rsc='O') and (-co2<co1)) then co1:=-co2;
if ((rsc='O') and (co1<-co2/2)) then begin co2:=-co1;nh:=1;end;
if ((rsc='O') and (co1<-co2)) then co2:=-co1;
```

D'abord, la boucle: for e:=1 to care do..., détermine les valeurs extrêmes de l'abscisse et de l'ordonnée sur les ensembles à représenter. Ces valeurs sont corrigées par un changement d'orientation éventuel. Les nombres {470, 270} sont les dimensions de la fenêtre, en pixels. Il faut prendre garde que, pour le système, l'écran est orienté de gauche à droite et de haut en bas. Le calcul du coefficient d'échelle verticale, co2, tient compte du nombre, nh (=1 ou 2), des écrans à superposer. Si rsc='O', on impose deux échelles égales en valeur absolue; et, si l'amplitude verticale est trop faible, on impose nh:=1.

1.2.4.4 Préparation d'une fenêtre rectangulaire

```
for hi:=1 to nh do begin
 if (nh=2) then if (hi=1) then begin
     setrect(recd,495,271,500,280);paintrect(recd) end
    else begin
     setrect(recd,495,1,500,10);paintrect(recd) end;
 moveto(20,293);
 if (nh=2) then if (hi=1) then drawstring('haut; ')
    else drawstring('bas;');
 drawstring(concat('ah=',chh,chr(48+ah),';av=',chv,chr(48+av),'; '));
 drawstring(concat(nomen,' '));
 Pch:=2-round(col*gch);Pcv:=10+round(co2*hau)+((1-hi)*270);
 moveto(Pch,Pcv);drawstring('+');
 moveto(470,Pcv);mot:=concat('axe',chh,chr(48+ah));drawstring(mot);
 moveto(Pch,12);mot:=concat('axe',chv,chr(48+av));drawstring(mot);
```

Selon que nh vaut 1 ou 2, il y a, pour un seul plan demandé, un ou deux affichages à l'écran: d'où la boucle: for hi:=1 to nh do...

Les instructions: setrect...paintrect..., jouant quand il y a deux écrans, marquent des petits rectangles noirs, respectivement au coin inférieur droit de l'écran supérieur et au coin supérieur droit de l'écran inférieur: pour assembler les deux images saisies, il faut superposer exactement ces deux rectangles. De plus, les deux images sont distinguées par la mention: 'haut', ou: 'bas', écrite en bas de la fenêtre.

En bas également, on écrit, dans tous les cas, les rangs des facteurs afférents à chaque axe (ah, horizontal; av, vertical); puis le nom du fichier. Ainsi, l'utilisateur sait à quoi se rapportent les images qu'il a saisies.

Le caractère '+' est mis au point de concours des axes (sous la réserve que ce point soit dans les limites de la fenêtre qu'on dessine). Puis (sous la même réserve) on met, à droite, le sigle de l'axe horizontal, là où celui-ci rencontre le bord; et, de même, en haut, le sigle de l'axe vertical, là où il coupe le bord.

On notera que l'ordre 'moveto' (aller à) est sans cesse employé pour fixer les coordonnées du point de l'écran à partir duquel s'exécute un ordre d'écriture: drawstring.

1.2.4.5 Boucle pour les éléments d'un ensemble: lecture sur le fichier des facteurs et affichage à l'écran

Comme au §1.2.3, pour le calcul des bornes des facteurs, on doit, pour l'affichage des points, ouvrir les fichiers de facteurs, et les informations, lues, comme des entiers, sont comprises grâce à la structure 'ricla' (d'où l'en-tête: with riri do...).

Pour chaque ensemble, par les ordres: texfont... setsize..., on impose au système la fonte et la taille choisies; on retourne à 'Monaco 9' (fonte=4, taille=9) avant de fermer, par: close(firi), le fichier qu'on a fini de lire.

```
    with riri do begin
     for e:=1 to care do if (pre[e]=1) then begin
      textfont(fte[e]);textsize(sze[e]);
      nomf:=concat(nomen,sigler(ens[e]),'Fac.w');moveto(5,293);lir(ch1);
      reset(firı,nomf);
      for u:=1 to titab-1 do read(firı,c);
      read(firi,cari);read(firi,c);amax:=c-2;dir:=7+(2*amax);
      for u:=1 to 3*(amax+2) do read(firi,c);
      for i:=1 to cari do begin
       for u:=1 to dır do read(firı,ptrcl^[u]);
       Pch:=2+round(col*((sh*Fac[ah])-gch));
       Pcv:=10+round(co2*((sv*Fac[av])+hau))+((1-hi)*270);
        mot:=sıgler(sis);
       if (sge[e] in ['1'..'4']) then begin rg:=rng[e];
          if (length(mot)<rg) then rg:= length(mot);mot:=copy(mot,rg,1);end;
       if (sge[e]='O') then mot:=sye[e];
       if (Pcv<283) then begin moveto(Pch,Pcv);drawstring(mot);end;end;
      textfont(4);textsize(9);close(firi) end end;
    moveto(5,293);
    lir(ch1);eraserect(rec0);end;
   moveto(20,20);write('faut-il tracer un autre plan oui(O) ou non(N) ');
   readln(rpp);wrıteln end;
  write('faut ıl poursuivre avec d''autres ensembles oui(O) ou non(N) ');
  readln(rpe) end;end;end;
end.
```

On notera que les ordres de lecture d'un caractère: lir(ch1), qui commandent le début de l'affichage d'un ensemble, ou l'effacement de l'écran, utilisent la procédure 'lir', décrite dans IA§1.3.5. Nous rappelons qu'en TML2 (le compilateur PASCAL utilisé avec les sources publiées ici), la commande de création d'un fichier graphique ne peut s'exécuter quand le programme a envoyé un ordre 'readln' de lecture à l'écran; d'où la nécessité d'un ordre tel que 'lir'.

Les éléments sont affichés par un ordre 'drawstring', après un 'moveto', vers un point dont les coordonnées (sur l'écran gradué en pixels) sont calculées, en fonction des facteurs, par des fomules où figurent les coefficients d'échelle, co1 et co2; les signes d'orientation des axes, sh et sv; et les bornes, gch et hau, qui (cf. §1.2.4.3) sont, en bref, les valeurs des facteurs pour l'élément extrême du nuage, situé au coin supérieur gauche du plan. Le mot à écrire peut être le sigle lui-même: sigler(sis), ou une lettre fixée par le choix de l'utilisateurs (cf. §1.2.2).

Quand ont été affichés, pour un plan, tous les ensembles spécifiés dans la série en cours (cf. §1.2.4.1), le programme demande s'il faut tracer un autre plan (dans cette même série); et si la réponse a été rpp='N', le programme sort de la boucle de la série en demandant s'il faut poursuivre avec d'autres ensembles; i.e. spécifier une nouvelle série. Si la réponse: rpe='N', se termine la boucle des séries (ouverte par: while not(rpe='N')..., cf. §1.2.4.1); et aussi la boucle: if (rpc='O') then begin... (cf. §1.2.3), qui comprend tout le jeu du programme 'plan', à condition que le choix des ensembles n'ait pas abouti à un échec. C'est ici la fin de la procédure 'plan' elle-même; et aussi celle de l'unité 'uplan.p': voilà pourquoi le mot 'end' est écrit quatre fois.

2 Création de graphiques sur un listage en format de texte: le programme 'PlanX'

À la différence de 'PlanF' le programme 'PlanX', dans la version que nous publions, est écrit d'une seule pièce; avec, après les déclarations, une seule procédure auxiliaire: 'ecrire'. À ceci près, il y a un exact parallélisme entre les deux programmes, qui ont pour données des ensembles d'éléments décrits par les mêmes fichiers de facteurs. Dans l'exposé, nous suivrons ce parallélisme; et n'entrerons dans le détail des instructions que quand 'PlanX' se distingue de 'PlanF'.

2.1 Les déclarations de 'PlanX' et la procédure 'ecrire'

```
program planX;
uses memtypes,quickdraw,osintf,toolintf,sane,uver;
const afmax=10;emax=20;lmn=50;lmx=136;
var  riri:ricla;firi:file of integer;ft:text;ptrcl:ptrc;
  chl,rpp,rpe,che,rpc,rep,rpv,rpl:char;
  nomf,nomen,lign,mot:string;
  pl0:array[1..127] of stringptr;li0:string;
  a,ah,av,i,cari,e,care,carde,Pch,Pcv,larch,hauli,ecr0,mq0,lm0,cc,c,
  hliecr,hliimp,amax,fmax,dir,erl,u:integer;
  res,col,co2,tav,tah,lav,lah:real;
  ens:array[1..emax] of zigle;pre:array[1..emax] of integer;
  sge,sye:array[1..emax]of char;
  inf,sup:array[1..emax,1..afmax] of single;
  gch,hau,drt,bas:single;
procedure ecrire(sss:string;hh,vv:integer);var n0,i0:integer;si0,si1:string;
  begin n0:=length(sss);si0:='';for i0:=1 to n0 do si0:=concat(si0,' ');
  si1:=copy(pl0[vv]^,hh,n0);mq0:=0;
  if (si0=si1) then begin delete(pl0[vv]^,hh,n0);insert(sss,pl0[vv]^,hh) end
      else begin mq0:=1;ecr0:=ecr0+1;lm0:=lm0+5 end; end;
```

Le lecteur reconnaîtra ici ce qu'il a déjà vu au §1.2.1.

Mais, aux informations relatives à la fenêtre, se substituent celles afférentes au bloc de texte où doit s'inscrire un graphique. La longueur des lignes (i.e. la largeur du plan en caractères) doit être comprise entre les deux bornes 'lmi' et 'lmax' (celle-ci fixée pour une utilisation de l'imprimante 'imagewriter'). La hauteur du plan, est, de même, fixée par un nombre de lignes 'hauli': comme ces lignes sont écrites simultanément, au fur et à mesure de la lecture des éléments (chacun avec son sigle et ses coordonnées), le plan est créé en mémoire centrale avant d'être écrit sur le listage (, destiné à être ouvert par un traitement de texte; et, éventuellement, imprimé). L'espace en mémoire centrale est réservé par le tableau de pointeurs 'pl0': on voit qu'ici, le nombre des lignes 'hauli' (i.e. la hauteur d'un plan) est limitée à 127 (nombre qu'on pourrait augmenter).

Corrélativement, aux ordres d'écriture à l'écran, 'moveto', 'drawstring', se substitue la procédure 'ecrire' qui, dans le texte afférent au plan, insère le sigle 'sss' dans la ligne numérotée 'vv', à partir du caractère de rang 'hh': {vv, hh} jouent ici le rôle de coordonnées. De plus, l'insertion n'est faite que si un espace blanc est libre; et, en cas d'échec, la variable 'manque', mq0, est mise à 1; et le compte des manquants ecr0 est augmenté; la place est calculée pour en écrire les sigles sur une ligne qui prend la longueur lm0 (cf. *infra*, §2.7.2).

2.2 Désignation du fichier de base

```
begin rep:='N';erl:=0;ptrcl:=ptrc(@rıri);fmax:=afmax;benzecri;
while not ((rep='O') or (erl=6)) do begin rep:='O';
 write('le fıchıer de base est ');readln(nomen);
 if not(setvol(@nomen,10)=noerr) then begin rep:='N';erl:=erl+1;
     writeln('ERREUR le disque manque ') end;
 if (rep='O') then begin
   for c:=1 to 2 do begin chl:=chr(104+c);
      nomf:=concat(nomen,chl,'Fac.w');
      if not(verif(@nomf)=2) then
        writeln('NB il n''y a pas de fichier des Facteurs pour',chl:2) end;
   write('le nom du fichıer de base est-ıl confirmé oui(O) ou non(N) ');
   readln(rep) end end;
rpe:='O';rpp:='O';rpc:='N';
```

L'instruction composée: while not ((rep='O') or (erl=6)) do..., a pour fonction de fixer le nom du fichier de base. Elle ne diffère de celle du programme principal de 'PlanF' (cf. §1.2) qu'en ce que, dans 'PlanF', le choix du fichier de base est, avec tous les tracés subséquents, inséré dans une boucle; ce qui permet de reprendre avec un autre fichier de base sans sortir du programme. Dans la version publiée ici de 'PlanX', au contraire, on ne peut considérer qu'un fichier principal unique; à moins de fermer puis rouvrir 'PlanX'.

2.3 Choix des ensembles susceptibles de figurer sur les graphiques

```
while not ((rpc='O') or (erl=6)) do begin
  write('le nombre (1 a',emax:3,') des ensembles considérés est ');
  readln(care);if (care<1) then care:=1;if (emax<care) then care:=emax;
  e:=0;rpc:='O';
  while (e<care) and (rpc='O') do begin e:=e+1;
    write('le sigle de l''ensemble no',e:2,' est ');
    readln(lıgn);if (4<length(lign)) then lign:=copy(lıgn,1,4);
    ens[e]:=zigler(lign);
    nomf:=concat(nomen,lign,'Fac.w');
    if not(verif(@nomf)=2) then begin erl:=erl+1;
       writeln('ERREUR le fichier des facteurs manque pour ',lign);
       if not(erl=6) then begin
         write('pour reprendre après cette erreur entrer une lettre ');
         readln(rpc) end;
       rpc:='N' end end;
  if (rpc='O') then begin
    write('ce choix est-ıl confirmé oui(O) ou non(N) ');
    readln(rpc) end end;
```

Dans 'PlanX', les ensembles susceptibles de figurer sur les plans, sont choisis dans une boucle: while not..., tout analogue à celle de la procédure 'plan', de 'PlanF', expliquée au §1.2.2. À une importante différence près: dans 'PlanF', en même temps qu'on choisit un ensemble: ens[e], on fixe, une fois pour toutes, la police de caractères et le mode de représentation (par le sigle ou par une seule lettre), propres aux éléments de ens[e]. Dans 'PlanX', il n'y a pas lieu de fixer une police, car le listage créé peut s'afficher correctement avec toutes polices (pourvu que les caractères aient tous même largeur); mais quant à l'écriture du sigle ou d'une lettre, le choix est offert à l'utilisateur, au seuil de chaque série de graphiques.

2.4 Calcul des bornes des facteurs sur chacun des ensembles

```
if (rpc='O') then begin
 writeln('afin de préparer le tracé on calcule les bornes des Facteurs');
 with riri do begin for e:=1 to care do begin
    for a:=1 to afmax do begin inf[e,a]:=100000;sup[e,a]:=-100000 end;
    nomf:=concat(nomen,sigler(ens[e]),'Fac.w');reset(firi,nomf);
    for u:=1 to titab-1 do read(firi,c);
    read(firi,cari);read(firi,c);amax:=c-2;dir:=7+(2*amax);
    if (amax<fmax) then fmax:=amax;
    for u:=1 to 3*(amax+2) do read(firi,c);
    for i:=1 to cari do begin
     for u:=1 to dir do read(firi,ptrcl^[u]);
     for a:=1 to fmax do begin
       if (Fac[a]<inf[e,a]) then inf[e,a]:=Fac[a];
       if (Fac[a]>sup[e,a]) then sup[e,a]:=Fac[a] end end;
    close(firi) end end;
```

Le calcul des bornes des facteurs sur les ensembles possible, se fait dans 'PlanX' exactement comme dans 'PlanF' (cf. *supra*, §1.2.3).

```
nomf:=concat(nomen,'plantx');rewrite(ft,nomf);writeln(ft,nomf);
for i:=1 to 127 do new(pl0[i]);
```

Après ce calcul, le fichier de texte, ou listage, destiné à recevoir les graphiques est créé (sur disque) avec le suffixe 'plantx'; et l'on réserve en mémoire centrale les 127 lignes sur lesquelles les graphiques plans sont composés, avant d'être écrits sur le listage 'plantx'.

2.5 Spécification d'une série de graphiques

```
while not(rpe='N') do begin carde:=0;
 writeln('choix des ensembles à représenter parmi ceux considérés');
 for e:=1 to care do begin pre[e]:=0;
  write('faut-il représenter ',sigler(ens[e]),' oui(O) ou non(N) ');
  readln(che);
  if not (che='N') then begin pre[e]:=1;sge[e]:='N';
    writeln('faut-il figurer les élém',sigler(ens[e]),' par leurs sigles(S)');
    write('ou par un caractère unique(U) ');readln(rpl);
    if (rpl='U') then begin sge[e]:='O';
        write('le caractère choisi pour ',sigler(ens[e]),' est ');
     readln(sye[e]) end end;
  carde:=carde+pre[e] end;
 if (carde=0) then begin pre[1]:=1;sge[1]:='N' end;rpp:='O';
```

Les ensembles à représenter dans une série sont choisis, ici, comme dans 'PlanF' (cf. §1.2.4.1); à ceci près que, comme on l'a annoncé au §2.3, le mode de représentation des éléments de chacun des ensembles, retenus pour une série, est fixé en tête de celle-ci. Toutefois, selon la version de 'PlanX' publiée ici, les éléments d'un ensemble ens[e] sont figurés, soit par leur sigle, soit par un caractère unique affecté à ens[e]; il n'est pas prévu de prendre, dans le sigle, un seul caractère de rang déterminé (cf. §1.2.2).

Si l'utilisateur n'a demandé, pour une série, aucun des ensembles possibles, le premiers de ceux-ci, ens[1], est imposé par défaut; les éléments étant figurés par leurs sigles (sge[1]:='N').

2.6 Spécification d'un graphique au sein d'une série

2.6.1 Choix des axes

```
while not(rpp='N') do begin
 writeln('choix des axes du plan');
 write('le numéro de l''axe horizontal est ah = ');
 readln(ah);if (ah<1) then ah:=1;if (fmax<ah) then ah:=fmax;
 write('le numéro de l''axe vertical est av = ');
 readln(av);if (av<1) then av:=1;if (fmax<av) then av:=fmax;
```

Comme au §1.2.4.2, l'utilisateur fixe d'abord les numéros des axes horizontal et vertical.

2.6.2 Calcul des bornes de l'abscisse et de l'ordonnée

```
gch:=0;hau:=0;drt:=0;bas:=0;cc:=0;
for e:=1 to care do if (pre[e]=1) then begin
  if (inf[e,ah]<gch) then gch:=inf[e,ah];
  if (sup[e,ah]>drt) then drt:=sup[e,ah];
  if (-sup[e,av]<hau) then hau:=-sup[e,av];
  if (-inf[e,av]>bas) then bas:=-inf[e,av] end;
```

Immédiatement, le programme calcule pour la réunion des ensembles représentés dans la série, les valeurs extrêmes des deux coordonnées: {haut, bas, gach, drt} (cf. §1.2.4.3, pour 'PlanF').

2.6.3 Choix des dimensions du graphique plan

```
writeln('choix des dimensions du graphique');
write('largeur du graphique en caractères = ');readln(larch);
if (larch<lmn) then larch:=lmn;if (lmx<larch) then larch:=lmx;
writeln('largeur = ',larch:4);
if (drt=gch) then co1:=0 else co1:=(larch-4)/(drt-gch);
res:=co1*(bas-hau);
hliecr:=1+round(res*(1/2));  hliimp:=1+round(res*(6/17));
writeln('afin d''avoir même échelle pour les deux axes');
writeln('sur l''écran demander une hauteur de',hliecr:4,' lignes');
writeln('sur le listage demander une hauteur de',hliimp:4,' lignes');
write('hauteur du graphique en lignes = ');readln(hauli);
if (hauli<10) then hauli:=10;if (127<hauli) then hauli:=127;
```

L'utilisateur fixe alors les dimensions du graphique. D'abord la largeur, 'larch' (laquelle, cf. §2.1, doit être comprise entre les deux bornes 'lmi' et 'lmax'). Puis le programme propose deux valeurs pour la hauteur ('hauli' = nombre de lignes) en tenant compte de l'impression usuelle par 'Imagewriter' ou de l'affichage, par le traitement de texte 'Edit', en 'Monaco 9'...

2.7 Composition d'un graphique plan

```
for i:=1 to larch do write(ft,'=');writeln(ft);
li0:='';for i:=1 to larch do li0:=concat(li0,' ');
for i:=1 to hauli do pl0[i]^:=li0;
if (bas=hau) then co2:=0 else co2:=-(hauli-1)/(bas-hau);
write(ft,'ensemble(s) représenté(s) : ');
for e:=1 to care do if (pre[e]=1) then write(ft,sigler(ens[e]):5);writeln(ft);
```

Le programme écrit d'abord, sur le listage, une ligne de signes '=' destinés à séparer le plan de tout ce qui le précède éventuellement. Le tableau des 'hauli' lignes de longueur 'larch' est mis en blanc dans la mémoire centrale. On écrit sur le listage la liste des ensembles représentés (dans la série en cours).

Commence alors une boucle, où les fichiers 'Fac.w' des ensembles à représenter sont lus comme des entiers, et compris sous le format 'ricla'

2.7.1 Écriture de l'en-tête du plan en lisant le premier ensemble représenté

```
with riri do begin
for e:=1 to care do if (pre[e]=1) then begin ecr0:=0;cc:=cc+1;
  nomf:=concat(nomen,sigler(ens[e]),'Fac.w');reset(firi,nomf);
  for u:=1 to 61 do read(firi,c);
  for u:=1 to ddir do read(firi,ptrcl^[u]);
  for u:=62+ddir to titab-1 do read(firi,c);
  read(firi,cari);read(firi,c);amax:=c-2;dir:=7+(2*amax);
  for u:=1 to 3*(amax+2) do read(firi,c);
  if (cc=1) then begin
   lah:=Fac[ah];tah:=lah/Fac[amax+1];
   lav:=Fac[av];tav:=lav/Fac[amax+1];
   write(ft,'axe horizontal:',ah:2,' ; min=',-(-gch):3,' ; max=',-(-drt):3);
   writeln(ft,' ; lam=',lah:3,' ;taux=',tah:3);
   write(ft,'axe vertical  :',av:2,' ; min=',-bas:3,' ; max=',-hau:3);
   writeln(ft,' ; lam=',lav:3,' ;taux=',tav:3) end;
```

Sont écrits sur deux lignes consécutives, respectivement pour l'axe horieontal et l'axe vertical, le rang du facteur, la valeur propre et le taux d'inertie. Compte tenu de la structure de l'en-tête du fichier 'Fac.w', (cf. IB§2.1.3) ces informations ont pu être comprises sous le format 'ricla'.

2.7.2 Traitement des éléments d'un ensemble représenté

```
write(ft,'élément(s) ',sigler(ens[e]),' non représenté(s) : ');lm0:=35;
for i:=1 to cari do begin
  for u:=1 to dir do read(firi,ptrcl^[u]);
  Pch:=1+round(col*(Fac[ah]-gch));Pcv:=1+round(co2*(Fac[av]+hau));
  mot:=sigler(sis);if (sge[e]='O') then mot:=sye[e];
  ecrire(mot,Pch,Pcv);
  if (sge[e]='N') then begin
     if (larch<lm0) then begin writeln(ft);lm0:=5 end;
     if (mq0=1) then write(ft,mot:5) end end;
close(firi);
writeln('il y a ',ecr0,' élément(s) ',sigler(ens[e]),' non représenté(s) : ')
if ((ecr0=0) or (sge[e]='O'))
   then writeln(ft,ecr0:4) else writeln(ft) end end;
```

Pour chaque ensemble représenté, est prévu, sur le listage, une ligne, ou un bloc de lignes, recensant les éléments qui n'ont pu être écrits, faute de place (cf. *supra*, procédure 'ecrire', §2.1). Si les éléments de ens[e] sont figurés par une seule lettre, seul est écrit le nombre des manquants; sinon, tous leurs sigles sont écrits explicitement, sur le listage 'ft', au fur et à mesure qu'ils se rencontrent (le nombre 'lm0' signalant s'il y a lieu d'aller à la ligne).

Quant au graphique plan, il ne peut être immédiatement écrit sur 'ft'; mais il est composé, dans l'espace des lignes pl0[i]^ réservées en mémoire centrale, en y adjoignant élément après élément, par la procédure 'ecrire'; dans la mesure où l'espace blanc requis est disponible.

2.7.3 Fin de la composition d'un graphique plan; fin du programme

```
  Pch:=1+round(-co1*gch);Pcv:=1+round(co2*hau);
  ecrire('+',Pch,Pcv);
  mot:=concat('axe',chr(48+ah));ecrire(mot,larch-3,Pcv);
  mot:=concat('axe',chr(48+av));ecrire(mot,Pch,1);
  ecrire(chr(124),1,Pcv);for i:=2 to larch do ecrire('-',i,Pcv);
  for i:=1 to hauli do ecrire('|',Pch,i);    write(ft,'|');
  for i:=1 to larch-2 do write(ft,'-');writeln(ft,'|');
  for i:=1 to hauli do begin ecrire('|',1,i);ecrire('|',larch,i) end;
  for i:=1 to hauli do writeln(ft,pl0[i]^);write(ft,'|');
  for i:=1 to larch-2 do write(ft,'-');writeln(ft,'|');
  write('faut-il tracer un autre plan oui(O) ou non(N) ');
  readln(rpp);writeln end;
 write('faut-il poursuivre avec d''autres ensembles oui(O) ou non(N) ');
 readln(rpe) end;
for i:=1 to 127 do dispose(pl0[i]);close(ft);
end;end.
```

Les éléments de tous les ensembles à représenter ont été placés. Dans la mesure de la place disponible, on ajoute l'origine, '+'; les noms des axes; les axes eux-mêmes, tracés avec les tirets 'l' et '-'; enfin, le cadre, tracé de même.

La boucle afférente au tracé de graphiques à l'intérireur d'une série se termine par la question: faut-il tracer un autre plan... Celle afférente à la spécification des séries, par: faut-il poursuivre avec d'autres ensembles...

L'espace demandé pour y composer les plans est libéré.

Le listage 'planX.p' se termine par deux 'end'. Le premier, met fin à l'instruction composée: if (rpc='O') then begin..., où l'on entre si le choix des ensembles possibles s'est achevé avec succès. Le dernier 'end' marque la fin de tout le programme.

3 Appendice: Lecture et création de fichiers

Nous consacrons l'espace disponible, à la fin du présent chapitre, à un inventaire des fichiers lus ou créés par les programmes considérés dans la série: [SOURCES PASCAL I].

3.1 Analyse factorielle: programme 'qori'

Les fichiers lus par 'qori' sont, d'une part, le tableau même à analyser:

Disq:Dss:tabsuf ,

le suffixe: suf, pouvant être: {" (vide); 'yy'; '.z'; '.w'}; selon que le tableau est un texte (contenant des entiers: "; ou des réels: 'yy'); ou un fichier binaire (lu sous le format entier: '.z'; ou réel: '.w') ;

et, d'autre part, éventuellement, un ou deux fichiers de texte dont chacun spécifie un sous-ensemble à mettre en supplément:

Disq:Dss:tabisupx ; Disq:Dss:tabisupx .

Les fichiers créés par 'qori' sont:

Disq:Dss:tabipr ; Disq:Dss:tabjpr ;

fichiers d'entiers, en format binaire, enregistrant l'existence éventuelle d'éléments supplémentaires: ces fichiers sont créés même si tous les éléments sont gardés en principal);

le listage d'analyse factorielle (fichier de texte):

Disq:Dss:tabcortx ;

les fichiers de facteurs pour les ensembles principaux, Ip et Jp:

Disq:Dss:tabiFac.w ; Disq:Dss:tabjFac.w ;

et, éventuellement, si est non-vide un ensemble, Is ou Js, d'éléments mis en supplément, le fichier de facteurs pour cet ensemble:

Disq:Dss:tabisFac.w ; Disq:Dss:tabjsFac.w.

3.2 Adjonction de tableaux supplémentaires: programme 'qorlsup'

Les fichiers lus par 'qorlsup' (cf. ID) sont, d'une part, le tableau même à adjoindre en supplément:

Disq:Dss:tabsigsuf ;

le sigle: sig, étant terminé par: a ou b; et le suffixe de format, suf, ayant la valeur rappelée, ci-dessus, à propos de 'qori';

et, d'autre part, les fichiers créés lors de l'analyse du tableau principal:

{Disq:Dss:tabipr , tabjpr , tabiFac.w, tabjFac.w} ;

Les fichiers créés par 'qorlsup' sont: le texte du listage des résultats et le fichier des facteurs, en format binaire:

Disq:Dss:tabsigcorsutx ; Disq:Dss:tabsigFac.w .

3.3 Création de graphiques: programmes 'PlanF' et PlanX'

Les programmes, 'PlanF' et 'PlanX', de tracé de graphiques plans, utilisent, pour seule donnée, les fichier de facteurs: 'Fac.w', afférents aux ensembles à représenter. On sait que 'PlanF' ne crée aucun fichier, le système du Macintosh permettant à l'utilisateur de saisir les graphiques affichés à l'écran. Tandis que 'PlanX' crée un listage unique, quel que soit le nombre des ensembles considérés et des graphiques demandés; i.e., si le tableau de base est: Disq:Dss:tab, un listage: Disq:Dss:tabplantx.