

Le Graph Template Language (GTL) : le futur de SAS/GRAPH

Contrairement à tous les autres éléments du langage graphique de SAS, le GTL fonctionne en deux temps : d'abord on définit un modèle (template), éventuellement paramétré, qui sera stocké à un emplacement pérenne, éventuellement partagé ; puis on utilise ce template au cours d'une étape Data où on lira les données alimentant le graphique, en affectant des valeurs aux paramètres s'il y en a. Outre une esthétique de très haute qualité due à l'emploi systématique d'un driver Java indépendant du circuit habituel décrit au début de ce chapitre, le GTL permet d'obtenir des résultats très soignés pour la juxtaposition de plusieurs graphiques, reprenant et améliorant les possibilités de la procédure GREPLAY.

1. Procédure TEMPLATE, stockage de modèle et étape Data

La procédure TEMPLATE qui permet de définir un modèle sera également détaillée au chapitre 5 pour d'autres utilisations. La syntaxe qui suit ne concerne que le versant de cette procédure lié au GTL.

Cette syntaxe est celle de la version 9.1.3 où le GTL est expérimental. En version 9.2, il faut ajouter des instructions BEGINGRAPH et ENDGRAPH qui encadrent le bloc allant de LAYOUT à ENDLAYOUT. On trouvera l'aide de la syntaxe 9.2, encore plus riche que celle décrite ici, à cette adresse : http://support.sas.com/documentation/cdl/en/grstatgraph/60787/HTML/default/main_contents.htm.

```
PROC TEMPLATE ;
  DEFINE STATGRAPH nomModele < / STORE = bibliothèque.itemStore > ;
  < DYNAMIC paramètre1 < paramètre2 < ... > > ; >
  < MVAR macroVariable1 < macroVariable2 < ... > > ; >
  LAYOUT OVERLAY | GRIDDED | LATTICE < / option(s) > ;
  ...
ENDLAYOUT ;
END ;
RUN ; QUIT ;
```

On donne aux modèles graphiques des noms, généralement à deux niveaux ; cela n'a rien d'obligatoire, mais permet de s'organiser plus aisément, le premier niveau étant une sorte de dossier, et le second le nom à proprement parler du modèle. Les noms autorisés, tant pour les dossiers que pour les modèles, suivent les normes usuelles de SAS : au maximum 32 caractères, ne commençant pas par un chiffre, et composé de lettres non accentuées, de chiffres et du signe *underscore*.

Les instructions DYNAMIC et MVAR permettent d'indiquer la présence de paramètres associés au modèle. MVAR donne le nom de macro-variables dont la valeur sera récupérée à l'exécution du modèle, au cours de l'étape Data qui y fera appel. DYNAMIC indique des noms de paramètres (sorte de variables texte internes au modèle) dont l'étape Data devra proposer des valeurs.

L'instruction `LAYOUT` décrit quel motif sous-tend le modèle graphique ; ils sont de trois types, décrits ci-dessous. L'instruction `ENDLAYOUT` termine le motif tandis que `END` termine la définition du modèle (faisant écho à `DEFINE`).

Une fois le modèle stocké à l'exécution de la procédure `TEMPLATE`, on y fait appel par une étape Data. Le lieu de stockage d'un template est indiqué au chapitre 5, section 5.1.2.

```
DATA _NULL_ ;  
  SET bibliothèque.tableSAS ;  
  FILE PRINT ODS=(TEMPLATE="nomModele" < DYNAMIC=(paramètre1="valeur"  
                                                    < paramètre2="valeur"  
                                                    < ... > ) ) ;  
  
  PUT _ODS_ ;  
RUN ;
```

L'étape Data qui appelle le modèle doit être insérée dans une destination ODS compatible : HTML, RTF ou PDF. Dans l'instruction `FILE PRINT`, on indique quel est le modèle, et les valeurs d'éventuels paramètres. Les valeurs des paramètres sont toujours considérées comme du texte et encadrées de guillemets. L'instruction `PUT _ODS_ ;` est indispensable à la production du graphique. L'exemple 5 montrera que si certains paramètres ne prennent pas de valeur, ils sont considérés comme un texte blanc.

2. Les motifs (LAYOUT)

Il existe trois motifs disponibles en GTL, qui se présentent comme des tableaux : `OVERLAY` qui superpose plusieurs dessins dans une unique cellule, `GRIDDED` et `LATTICE` qui accueillent un ou plusieurs dessins dans des cellules organisées en lignes et en colonnes.

Dans un motif `LATTICE`, toutes les cellules du tableau dans lequel sont organisés les dessins sont de même taille ; dans un motif `GRIDDED`, ces cellules adaptent leur taille à leur contenu (ce qui est pratique pour mettre un texte au-dessus d'un graphique). De plus, le motif `GRIDDED` accepte des fusions de cellules sur plusieurs lignes et/ou plusieurs colonnes. On atteint alors la même souplesse que pour les modèles de la procédure `GREPLAY`.

Les principales options à ajouter aux motifs `GRIDDED` et `LATTICE` sont `ROWS` et `COLUMNS`, qui indiquent le nombre de lignes et de colonnes du motif.

Il est possible d'imbriquer plusieurs niveaux de motifs, sauf à l'intérieur d'un motif `OVERLAY` : voir à ce sujet la section 6.

Tous les dessins qui vont cohabiter dans un motif `OVERLAY` partagent leurs axes : ceux-ci sont donc définis, si besoin, au niveau du motif. Pour les autres motifs, ces options s'insèrent dans l'instruction de dessin elle-même, permettant à chaque dessin de gérer ses axes indépendamment.

Les motifs disponibles en SAS 9.2 sont encore plus nombreux, et contiennent notamment `DATALATTICE` et `DATAPANEL`, des motifs juxtaposés en quadrillage dont le nombre de cases dépend d'une ou de deux variables, et prend donc des valeurs dynamiques en fonction des données.

3. Les instructions de dessin

Les principales instructions de dessin sont résumées dans le tableau 1. On y distingue deux caractéristiques de ces instructions : soit elles attendent des données détaillées, soit des statistiques comme source de données : et par ailleurs, soit elles sont des dessins de base (utilisables dans tous les motifs, à raison d'un et un seul par cellule), soit elles sont des dessins à superposer (utilisables seulement dans un motif OVERLAY).

Tableau 1 – Principales instructions de dessin du GTL

Type de dessin	Sur les données détaillées		Sur les statistiques	
	Élément de base	Élément superposé	Élément de base	Élément superposé
Nuage de points	NEEDLEPLOT SCATTERPLOT SCATTERPLOTMATRIX	NEEDLE SCATTER		
Courbe	STEPLOT SERIESPLOT	STEP SERIES		LINEPARM
Intervalle de confiance	BANDPLOT	BAND ELLIPSE		ELLIPSEPARM
Diagramme en bâtons	HISTOGRAM	HISTO	BARCHARTPARM HISTOGRAMPARM	BARPARM HISTOPARM
Courbe de densité	DENSITYPLOT	DENSITY		
Boxplot	BOXPLOT	BOX	BOXPLOTPARM	BOXPARM
Graphique rayonnant	VECTORPLOT	VECTOR		

La syntaxe de quelques-unes de ces instructions est détaillée ci-dessous. Les options touchant à l'aspect des axes sont renvoyées à la section suivante, puisque selon le motif on les définit dans l'instruction LAYOUT ou dans celle de dessin.

Boxplot ou boîte à moustaches (principales options)

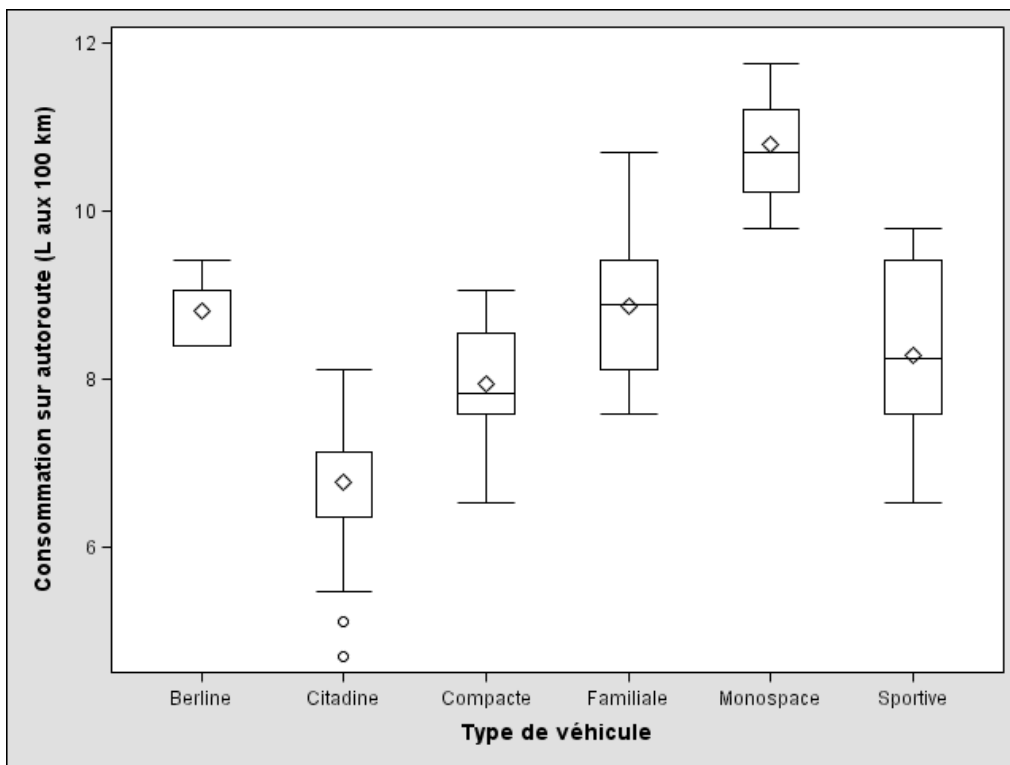
```
BOXPLOT Y=variableEtudiee < X=variableGroupeBoxplots > /
        DATALABEL=variableID LABELFAR=TRUE|FALSE ORIENT=HORIZONTAL|VERTICAL ;
```

Si on indique seulement une variable Y, il y aura seulement une boxplot. Si on ajoute une variable X, il y aura autant de boxplots que de valeurs de cette variable X. On peut ajouter, pour les points situés en dehors des barres de la boxplot, des labels (activés avec LABELFAR=TRUE) reprenant les valeurs d'une variable indiquée à l'option DATALABEL.

On obtient au choix des boxplots horizontales ou verticales selon l'option ORIENT (verticales par défaut).

Exemple 1 – Boxplots en GTL

```
PROC TEMPLATE ;
  DEFINE STATGRAPH exemples.boxplot ;
    DYNAMIC varY varGpe ;
    LAYOUT GRIDDED ;
      BOXPLOT Y=varY X=varGpe ;
    ENDLAYOUT ;
  END ;
RUN ;
DATA _NULL_ ;
  SET livre.voitures ;
  FILE PRINT ODS=(TEMPLATE="exemples.boxplot"
                  DYNAMIC=(varY="conso_auto"
                            varGpe="type")) ;
  PUT _ODS_ ;
RUN ;
```



Histogramme (principales options)

```
HISTOGRAM variableEtudiee / BINSTART=valeur
                          BINWIDTH=valeur
                          NBINS=nbBarres
                          FREQ=varPoids
                          ORIENT=HORIZONTAL|VERTICAL
```

SCALE=COUNT|PERCENT|PROPORTION ;

Les 3 premières options servent à forcer le découpage de la variable à étudier : BINSTART indique la borne inférieure du premier bâton, BINWIDTH la largeur (en terme d'intervalles) de chacun des bâtons, et NBINS leur nombre total. Si on souhaite par exemple avoir des intervalles 0-10, 10-20 et 20-30, on écrira BINSTART=0, BINWIDTH=10 et NBINS=3.

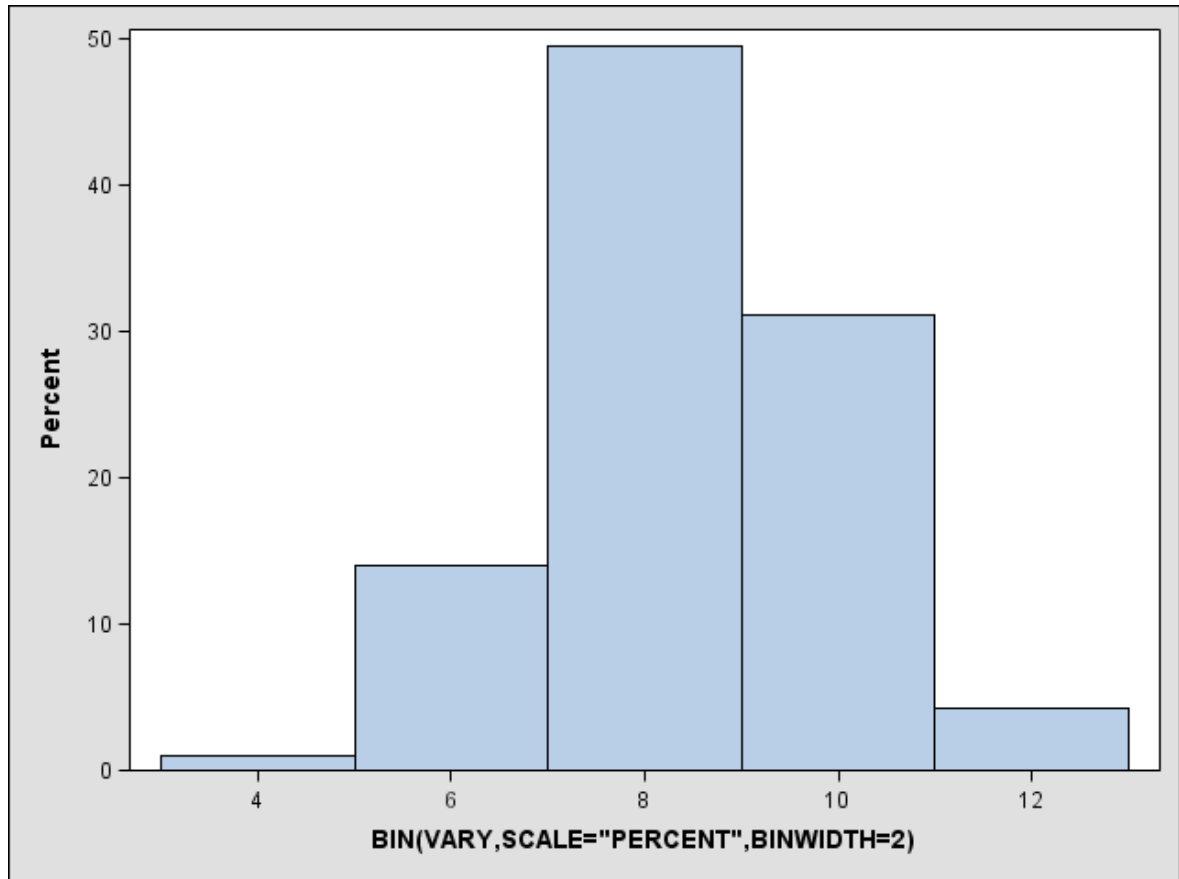
L'option FREQ permet d'indiquer le nom d'une variable servant de pondération : au lieu de dépendre du nombre d'observations, la taille des barres dépendra de la somme de cette variable.

L'option SCALE indique la graduation de l'axe correspondant à la taille des barres : en nombre d'observations (ou somme s'il y a une variable FREQ) avec la valeur COUNT, ou en pourcentage (entre 0 et 100 avec SCALE=PERCENT, entre 0 et 1 avec SCALE=PROPORTION).

L'option ORIENT est la même que pour l'instruction BOXPLOT.

Exemple 2 – Histogramme en GTL

```
PROC TEMPLATE ;
  DEFINE STATGRAPH exemples.histo ;
    DYNAMIC varY ;
    LAYOUT GRIDDED ;
      HISTOGRAM varY / SCALE=PERCENT BINWIDTH=2 ;
    ENDLAYOUT ;
  END ;
RUN ;
ODS HTML GPATH="c:\temp" ;
DATA _NULL_ ;
  SET livre.voitures ;
  FILE PRINT ODS=(TEMPLATE="exemples.histo" DYNAMIC=(varY="conso_auto")) ;
  PUT _ODS_ ;
RUN ;
ODS HTML CLOSE ;
```



Courbe de densité (principales options)

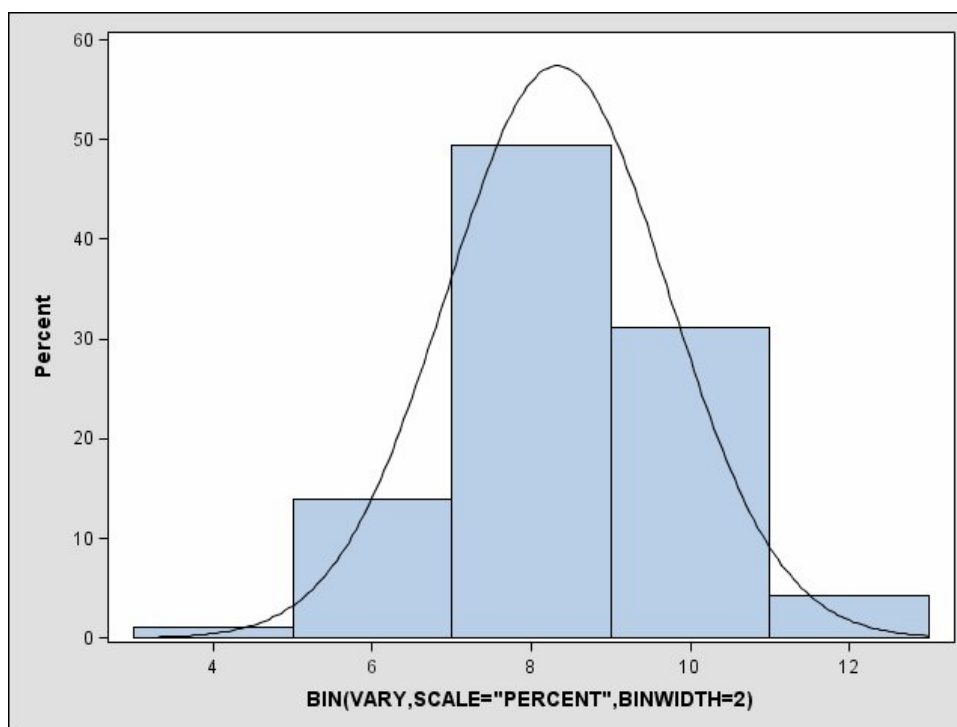
```
DENSITYPLOT variableEtudiee / KERNEL() | NORMAL() ;
```

On ajoute l'option `KERNEL()` pour dessiner une courbe de densité empirique. Si on précise plutôt l'option `NORMAL()`, ce sera la courbe de densité d'une loi normale de mêmes moyenne et variance.

Exemple 3 – Histogramme et courbe normale en GTL

```
PROC TEMPLATE ;  
  DEFINE STATGRAPH exemples.histoDensite ;  
  DYNAMIC varY ;  
  LAYOUT OVERLAY ;  
    HISTOGRAM varY / SCALE=PERCENT BINWIDTH=2 ;  
    DENSITY varY / NORMAL() ;  
  ENDLAYOUT ;  
END ;  
RUN ;  
ODS HTML GPATH="c:\temp" ;
```

```
DATA _NULL_ ;  
  SET livre.voitures ;  
  FILE PRINT ODS=(TEMPLATE="exemples.histoDensite"  
                 DYNAMIC=(varY="conso_auto")) ;  
  PUT _ODS_ ;  
RUN ;  
ODS HTML CLOSE ;
```











Nuage de points (principales options)

```
SCATTERPLOT Y=variableOrdonnees X=variableAbscisses  
            / DATALABEL=variableID  
            MARKERSYMBOL=symbole|CHARACTER  
            GROUP=variableBy ;
```

L'option DATALABEL est identique à celle de l'instruction BOXPLOT, les options GROUP et MARKERSYMBOL sont nouvelles. La première propose une série de points marquée différemment par valeur de cette variable (sur le modèle de l'instruction PLOT varY * varX = varBy dans la procédure GPLOT). Les noms des différents symboles qui peuvent être indiqués derrière l'option MARKERSYMBOL se trouvent résumés sur la figure 1. Si une variable est indiquée dans GROUP et que MARKERSYMBOL=CHARACTER, alors les points seront rendus par les valeurs de la variable de groupe.

Figure 1 – Quelques exemples de symboles utilisables en GTL

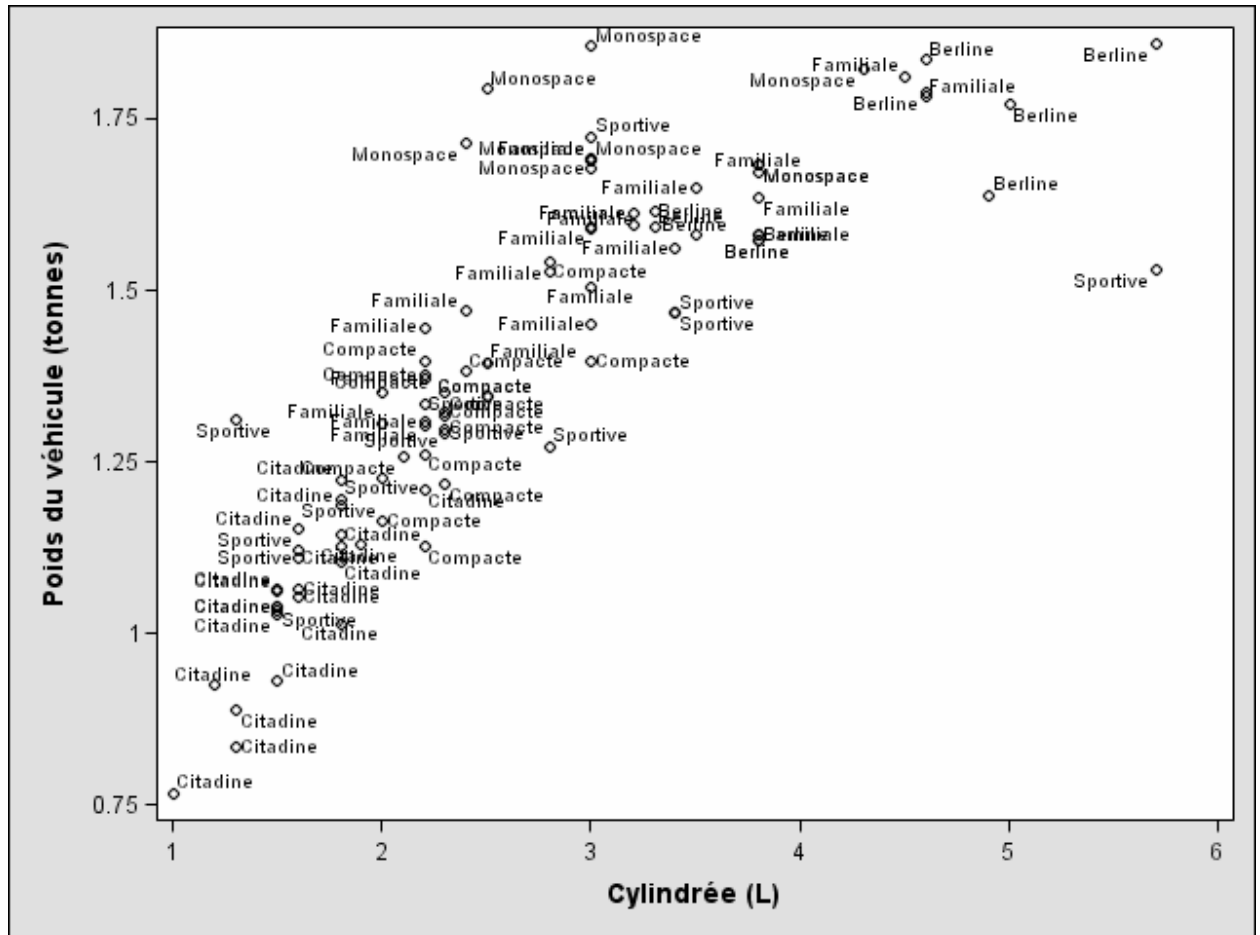
Nom	Symbole	Nom	Symbole
CIRCLE	○	FEMALEFIGUREFILLED	
CIRCLEFILLED	●	MALEFIGUREFILLED	
CROSS	+	MARSBOLD	
DIAMOND	◇	VENUSBOLD	
DIAMONDNARROW	◊	STAR	
FLAG	⚑	TRIANGLE	
PLUS	+	TRIANGLEDOWN	
SALTIRE	⊗	SQUAREEX	

Exemple 4 – Nuage de points en GTL

```

PROC TEMPLATE ;
  DEFINE STATGRAPH exemples.nuage ;
  DYNAMIC varX varY varId ;
  LAYOUT GRIDDED ;
    SCATTERPLOT X=varX Y=varY / DATALABEL=varId ;
  ENDLAYOUT ;
END ;
RUN ;
ODS HTML GPATH="c:\temp" ;
DATA _NULL_ ;
  SET livre.voitures ;
  FILE PRINT ODS=(TEMPLATE="exemples.nuage"
    DYNAMIC=(varY="poids" varX="cylindree" varId="type")) ;
  PUT _ODS_ ;
RUN ;
ODS HTML CLOSE ;

```



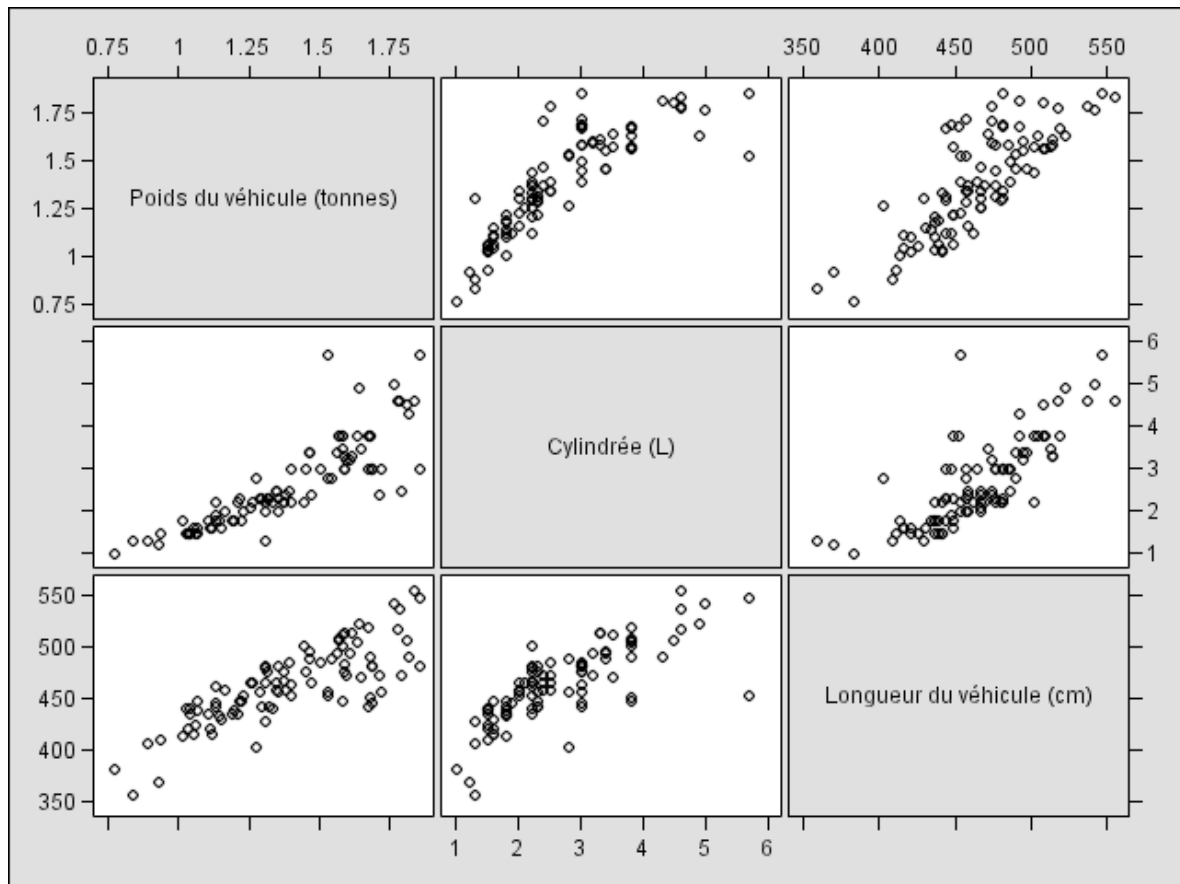
Matrice de nuages de points (principales options)

```
SCATTERPLOTMATRIX variable1 variable2 < variable 3 < ... > >  
/ DATALABEL=variableID  
GROUP=variableBy  
DIAGONAL=HISTOGRAM  
MARKERSYMBOL=symbole | CHARACTER ;
```

On énumère dans cette instruction une liste de noms de variables qui seront toutes croisées les unes avec les autres dans une organisation carrée. La diagonale de ce tableau de graphiques est vide par défaut, mais contient des histogrammes si on ajoute l'option `DIAGONAL=HISTOGRAM`.

Exemple 5 – Matrice de nuages de points en GTL

```
PROC TEMPLATE ;  
  DEFINE STATGRAPH exemples.nuageS ;  
  DYNAMIC var1 var2 var3 var4 ;  
  LAYOUT GRIDDED ;  
    SCATTERPLOTMATRIX var1 var2 var3 var4 ;  
  ENDLAYOUT ;  
END ;  
RUN ;  
ODS HTML GPATH="c:\temp" ;  
DATA _NULL_ ;  
  SET livre.voitures ;  
  FILE PRINT ODS=(TEMPLATE="exemples.nuageS"  
                 DYNAMIC=(var1="poids" var2="cylindree" var3="longueur")) ;  
  PUT _ODS_ ;  
RUN ;  
ODS HTML CLOSE ;
```



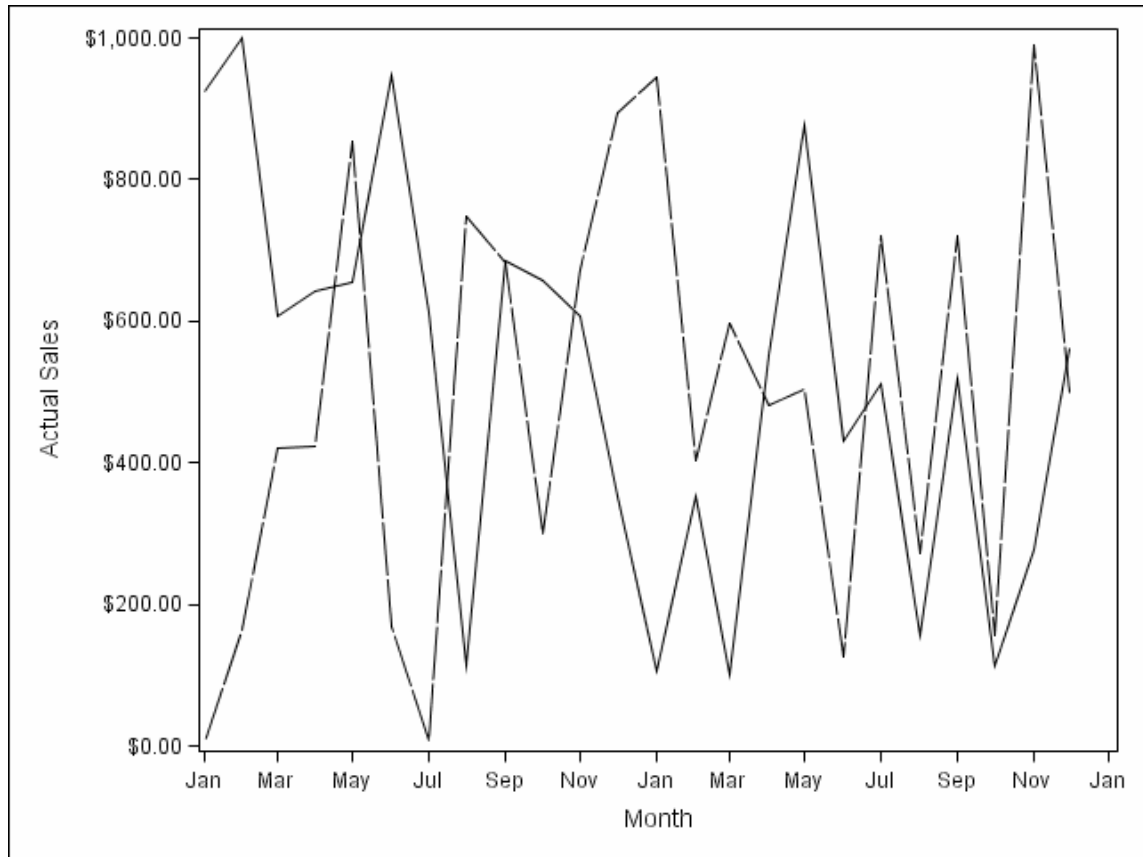
Courbe (principales options)

```
SERIESPLOT Y=variableOrdonnees X=variableAbscisses  
          / DATALABEL=variableID GROUP=variableBy  
          MARKERSYMBOL=symbole|CHARACTER  
          LINECOLOR=couleur LINETHICKNESS=dimension  
          LINEPATTERN=SOLID|DASH|DOT ;
```

En sus des options déjà connues, LINECOLOR, LINETHICKNESS et LINEPATTERN définissent l'aspect de la courbe. Pour la 1^{ère}, les codes couleurs habituels sont utilisés. Pour la 2^{nde}, l'épaisseur du trait est indiquée en pourcentage (2%), en points (2pt) ou en pixels (2px). Pour la dernière, les valeurs correspondent à un trait plein (SOLID), en pointillés longs (DASH) ou courts (DOT).

Exemple 6 – Courbe en GTL

```
PROC TEMPLATE ;  
  DEFINE STATGRAPH exemples.courbe ;  
  DYNAMIC varX varY varBy ;  
  LAYOUT GRIDDED ;  
    SERIESPLOT X=varX Y=varY / GROUP=varBy ;  
  ENDLAYOUT ;  
END ;  
RUN ;  
ODS HTML GPATH="c:\temp" STYLE=JOURNAL ;  
DATA _NULL_ ;  
  SET sashelp.prdsale (WHERE=(country="CANADA"  
                             AND product="SOFA" AND region="EAST")) ;  
  FILE PRINT ODS=(TEMPLATE="exemples.courbe"  
                 DYNAMIC=(varX="month" varY="actual" varBy="division")) ;  
  PUT _ODS_ ;  
RUN ;  
ODS HTML CLOSE ;
```



Droite (principales options)

```
LINEPARM YINTERCEPT|XINTERCEPT=valeur SLOPE=valeur  
          / LINECOLOR=couleur  
          LINETHICKNESS=dimension  
          LINEPATTERN=SOLID|DASH|DOT ;
```

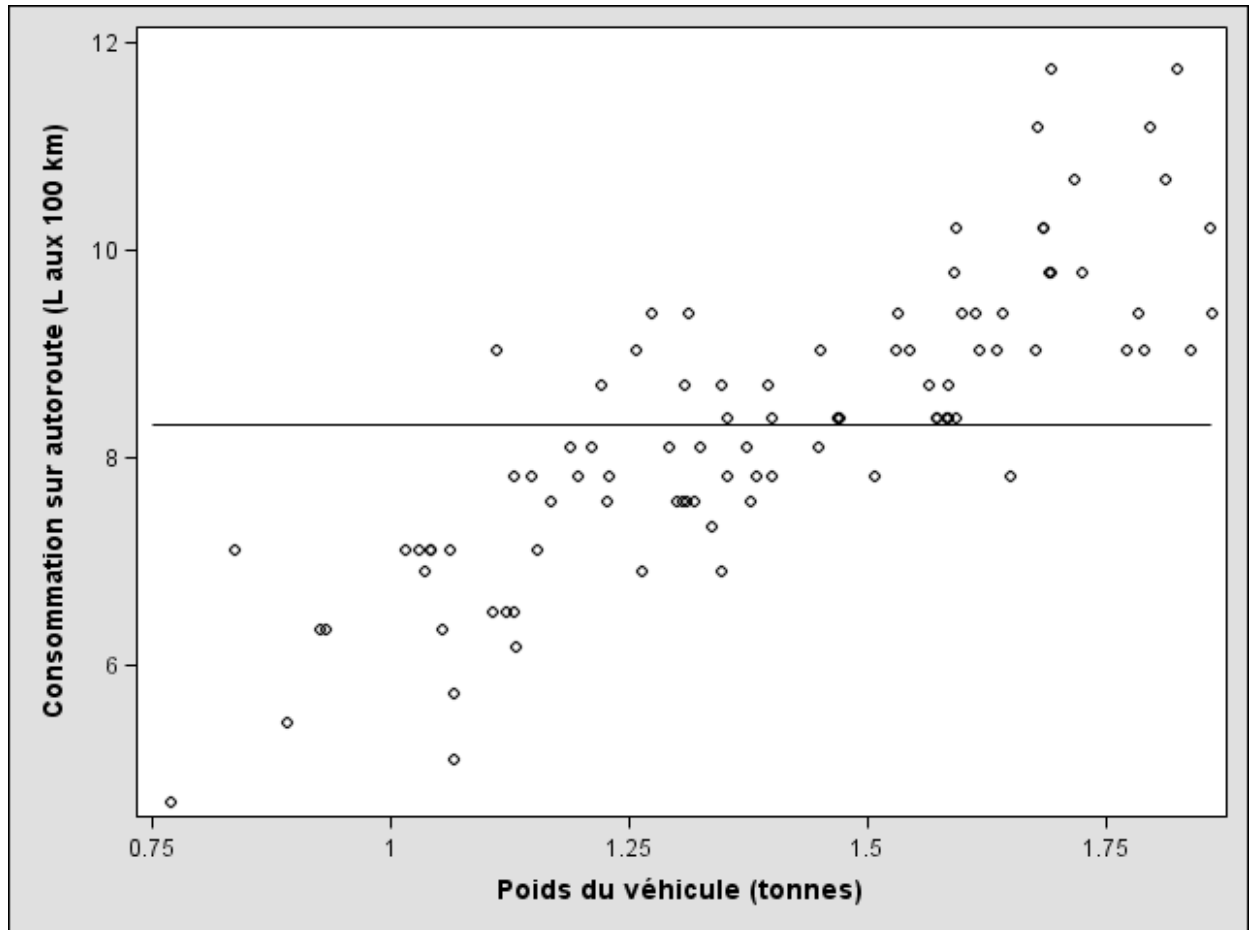
Les options YINTERCEPT et XINTERCEPT permettent d'indiquer en quel point la droite croise un axe. Si on indique uniquement XINTERCEPT, on trace une droite verticale ; si c'est YINTERCEPT seulement, une droite horizontale. On obtient une oblique avec YINTERCEPT et SLOPE (la pente) simultanément.

Exemple 7 – Nuage de points et droite superposés en GTL

```
PROC TEMPLATE ;
  DEFINE STATGRAPH exemples.nuage2 ;
  DYNAMIC varX varY ;
  LAYOUT OVERLAY ;
    SCATTERPLOT X=varX Y=varY ;
    LINEPARM YINTERCEPT=EVAL(MEAN(varY)) ;
  ENDLAYOUT ;
END ;
RUN ;
ODS HTML GPATH="c:\temp" ;
DATA _NULL_ ;
  SET livre.voitures ;
  FILE PRINT ODS=(TEMPLATE="exemples.nuage2"
                  DYNAMIC=(varX="poids" varY="conso_auto")) ;
  PUT _ODS_ ;
RUN ;
ODS HTML CLOSE ;
```

Dans l'exemple 7, on voit qu'on peut également proposer pour valeur d'une option un résultat de calcul. Pour cela, il faut l'inclure dans une instruction EVAL (la logique est semblable à celle de %EVAL en macro-langage : il faut indiquer au compilateur qu'il y a là un calcul à réaliser).

A l'intérieur de la fonction EVAL, on peut inclure les opérations de base (addition, soustraction, multiplication et division), ou invoquer des fonctions statistiques telles que MEAN, MIN, MAX, STD, MEDIAN, N, NMISS, ou encore les fonctions licites dans une condition WHERE (comme SUBSTR, ROUND, YEAR, etc.).



Graphique rayonnant (principales options)

```
VECTORPLOT Y=variableOrdonnees X=variableAbscisses  
           / DATALABEL=variableID GROUP=variableBy  
           ARROWDIRECTION=IN|OUT|BOTH|NONE  
           XORIGIN=valeur YORIGIN=valeur ;
```

Les options nouvelles ici sont ARROWDIRECTION pour indiquer comment sont tournées les flèches (OUT par défaut) et le couple XORIGIN / YORIGIN qui détermine de quel point du graphique partent les flèches : c'est le point de coordonnées (0;0) par défaut.

Le résultat d'un tel dessin est illustré à l'exemple 8.

4. La personnalisation des axes

Tableau 2 – Principales options agissant sur les axes

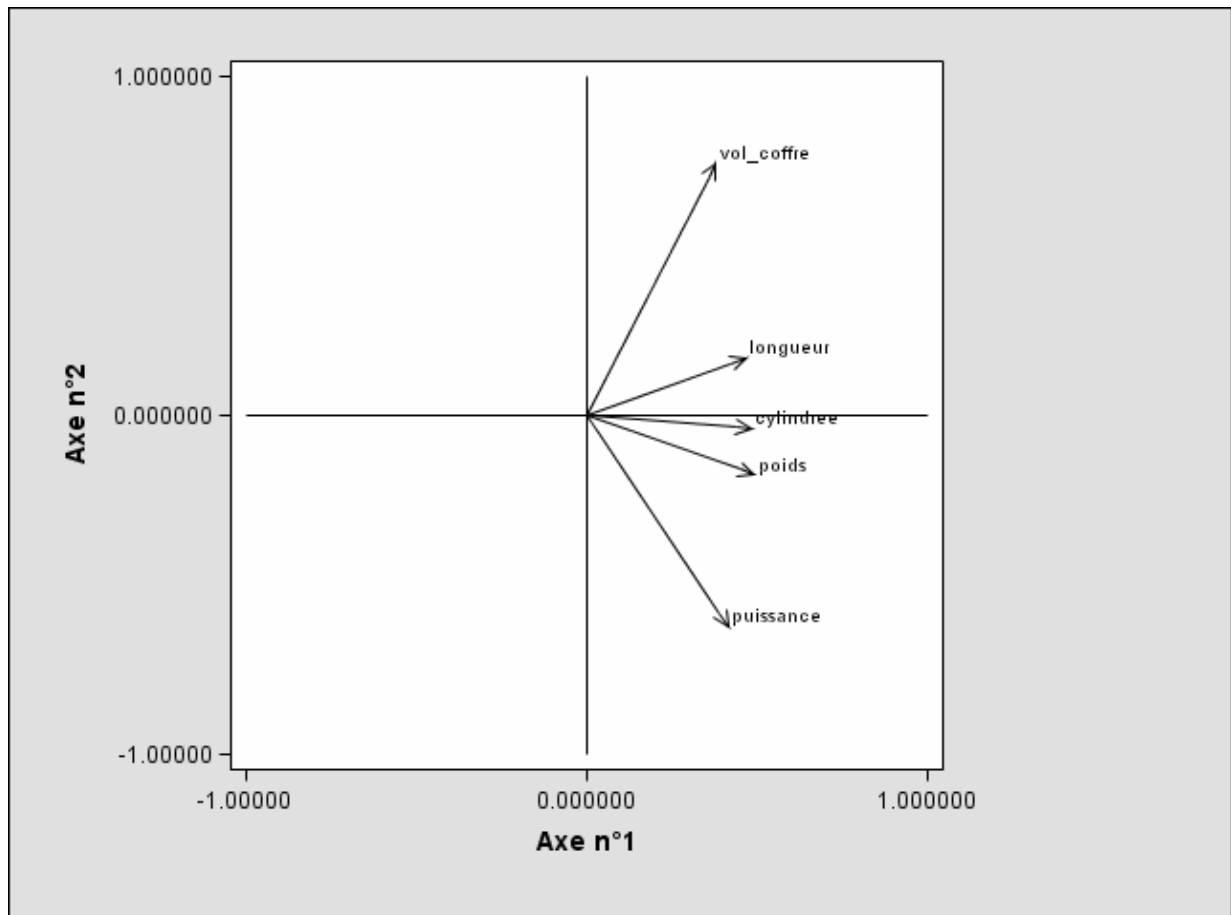
XGRID, YGRID=TRUE FALSE	Lignes de références verticales, horizontales
YEQUATED=TRUE FALSE	Graphique avec des échelles identiques en X et Y
XMIN, XMAX, YMIN, YMAX = valeur	Amplitude des axes
XAXIS=X X2, YAXIS=Y Y2	Axe affiché en bas/en haut, à gauche/à droite
XAXISOPTS, X2AXISOPTS, YAXISOPTS, Y2AXISOPTS = options	Spécifications des axes du bas, du haut, de gauche, de droite (valeurs : cf. infra)

Le tableau 2 résume les principales options du GTL pour personnaliser les axes d'un graphique. Les principales valeurs pour les options XAXISOPTS et consorts sont :

- DISPLAY=ALL|NONE| (élément(s)) où les éléments à afficher sont LABEL, LINE (=l'axe lui-même), TICKS (=les graduations) et VALUES ;
- TICKS=(valeurs) pour forcer les valeurs des graduations ;
- LABEL="intitulé" pour forcer le libellé de l'axe. En revanche, LABEL=NONE ne fonctionne pas ; il faut écrire LABEL=" " ou DISPLAY=(LINE TICKS VALUES).

Exemple 8 – Graphique d'une ACP (reprise de l'exemple 4.18) en GTL

```
PROC TEMPLATE ;
  DEFINE STATGRAPH exemples.acp ;
    LAYOUT OVERLAY / YEQUATED=TRUE
      XAXISOPTS=(TICKS=(-1 0 1) LABEL="Axe n°1")
      YAXISOPTS=(TICKS=(-1 0 1) LABEL="Axe n°2") ;
    VECTORPLOT X=prin1 Y=prin2 / DATALABEL=variable ;
    LINEPARM YINTERCEPT=0 ; LINEPARM XINTERCEPT=0 ;
  ENDLAYOUT ;
END ;
RUN ;
ODS OUTPUT eigenVectors = work.planFactoriel ;
PROC PRINCOMP DATA = livre.voitures ;
  VAR longueur poids cylindree puissance vol_coffre ;
RUN ;
ODS HTML GPATH="c:\temp" ;
DATA _NULL_ ;
  SET work.planFactoriel ;
  FILE PRINT ODS=(TEMPLATE="exemples.acp") ;
  PUT _ODS_ ;
RUN ;
ODS HTML CLOSE ;
```



5. L'insertion de textes

Construire une légende

Pour constituer la légende d'un graphique, on utilisera l'instruction DISCRETELEGEND. Mais avant cela, il aura fallu préparer le terrain : chaque élément graphique à faire figurer dans la légende recevra un nom (option NAME= dans l'instruction de dessin correspondante) et un texte à faire figurer dans la légende (option LEGENDLABEL=). Par exemple...

```
HISTOGRAM ... ;  
DENSITY ... / NORMAL() NAME="norm" LEGENDLABEL="Loi normale" ;  
DENSITY ... / KERNEL() NAME="noyau" LEGENDLABEL="Densité empirique" ;  
DISCRETELEGEND "norm" "noyau" ;
```

Après la liste des noms d'éléments figurant dans la légende, on trouve des options pour préciser l'allure de la légende.

```
DISCRETELEGEND noms / ACROSS=nombreColonnes DOWN=nombreLignes
```

```
BORDER=TRUE|FALSE TRANSPARENCY=nombre  
TITLE=texte  
HALIGN=LEFT|CENTER|RIGHT VALIGN=TOP|CENTER|BOTTOM ;
```

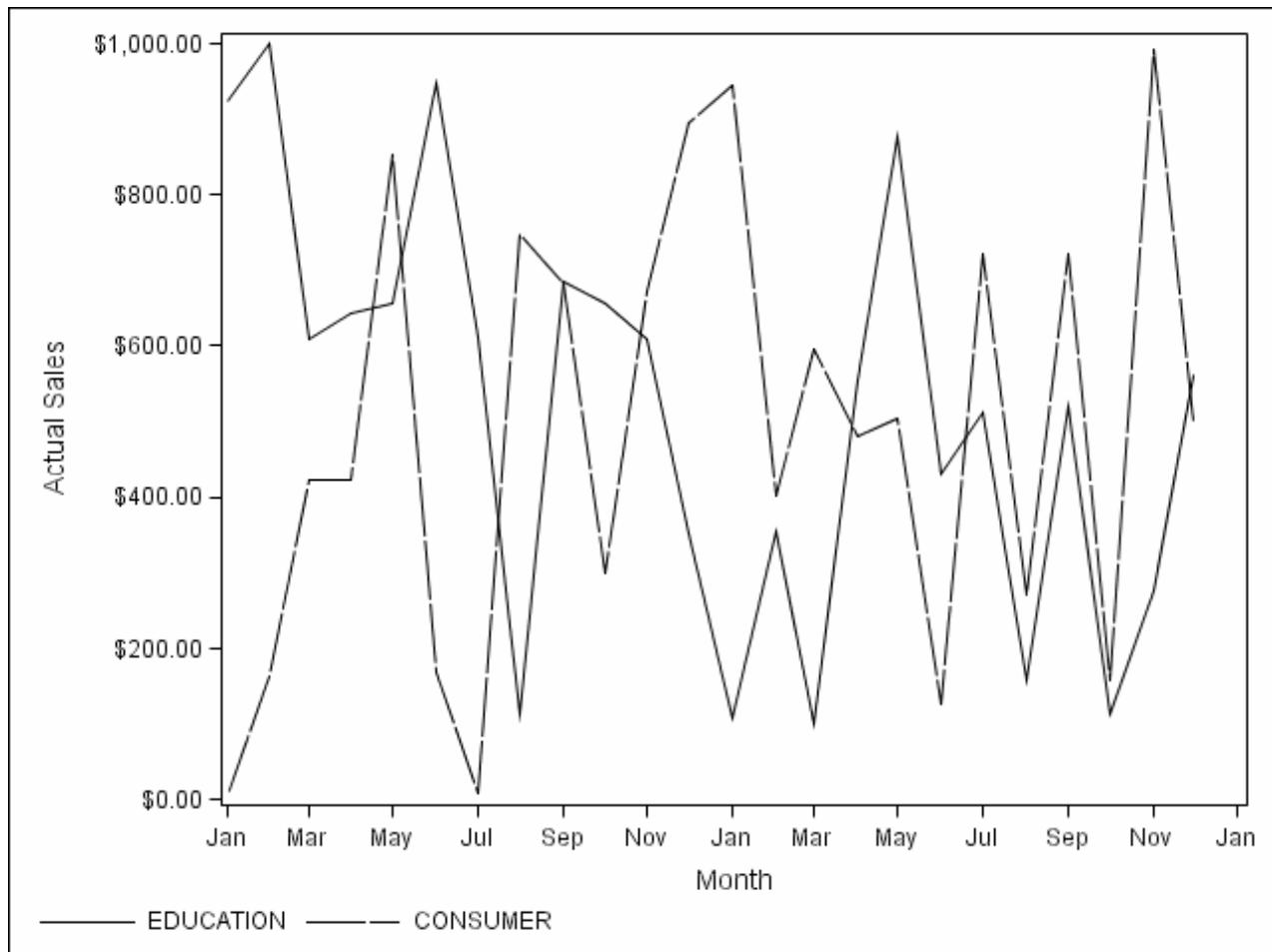
L'option `ACROSS` détermine le nombre d'éléments par ligne de légende, tandis que `DOWN` indique le nombre de lignes de la légende. Il est possible de n'indiquer que l'une de ces options ; si on précise les deux, s'assurer que le nombre d'entrées est inférieur au nombre de cases !

L'option `BORDER` dessine une bordure autour de la légende (par défaut, il n'y en a pas), `TITLE` indique le texte figurant en 1^{ère} ligne (non comptée dans le nombre `DOWN`). L'opacité de la légende est définie par l'option `TRANSPARENCY`, avec un nombre allant de 0 (opaque) à 1 (totalement transparent).

Enfin, `HALIGN` et `VALIGN` indiquent l'emplacement de la légende (par défaut, en plein centre du graphique !).

Exemple 9 – Courbe et légende en GTL (exemple 6 amélioré)

```
PROC TEMPLATE ;  
  DEFINE STATGRAPH exemples.courbe2 ;  
  DYNAMIC varX varY varBy ;  
  LAYOUT GRIDDED ;  
    SERIESPLOT X=varX Y=varY / GROUP=varBy NAME="courbe" ;  
    DISCRETELEGEND "courbe" / HALIGN=LEFT VALIGN=TOP ;  
  ENDLAYOUT ;  
END ;  
RUN ;  
ODS HTML GPATH="c:\temp" STYLE=JOURNAL ;  
DATA _NULL_ ;  
  SET sashelp.prdsale (WHERE=(country="CANADA"  
    AND product="SOFA" AND region="EAST")) ;  
  FILE PRINT ODS=(TEMPLATE="exemples.courbe2"  
    DYNAMIC=(varX="month" varY="actual" varBy="division")) ;  
  PUT _ODS_ ;  
RUN ;  
ODS HTML CLOSE ;
```



Ajouter titres et autres textes

On peut intégrer à un graphique des titres ainsi que du texte normal (la nuance n'étant apparente qu'en termes de styles : polices et couleurs utilisées). Pour cela, on dispose d'instruction `ENTRYTITLE` et `ENTRY`, respectivement. Une instruction `ENTRYFOOTNOTE` existe également, mais on peut s'en passer puisque les « titres » (`ENTRYTITLE`) peuvent être positionnés indifféremment en haut ou en bas du graphique.

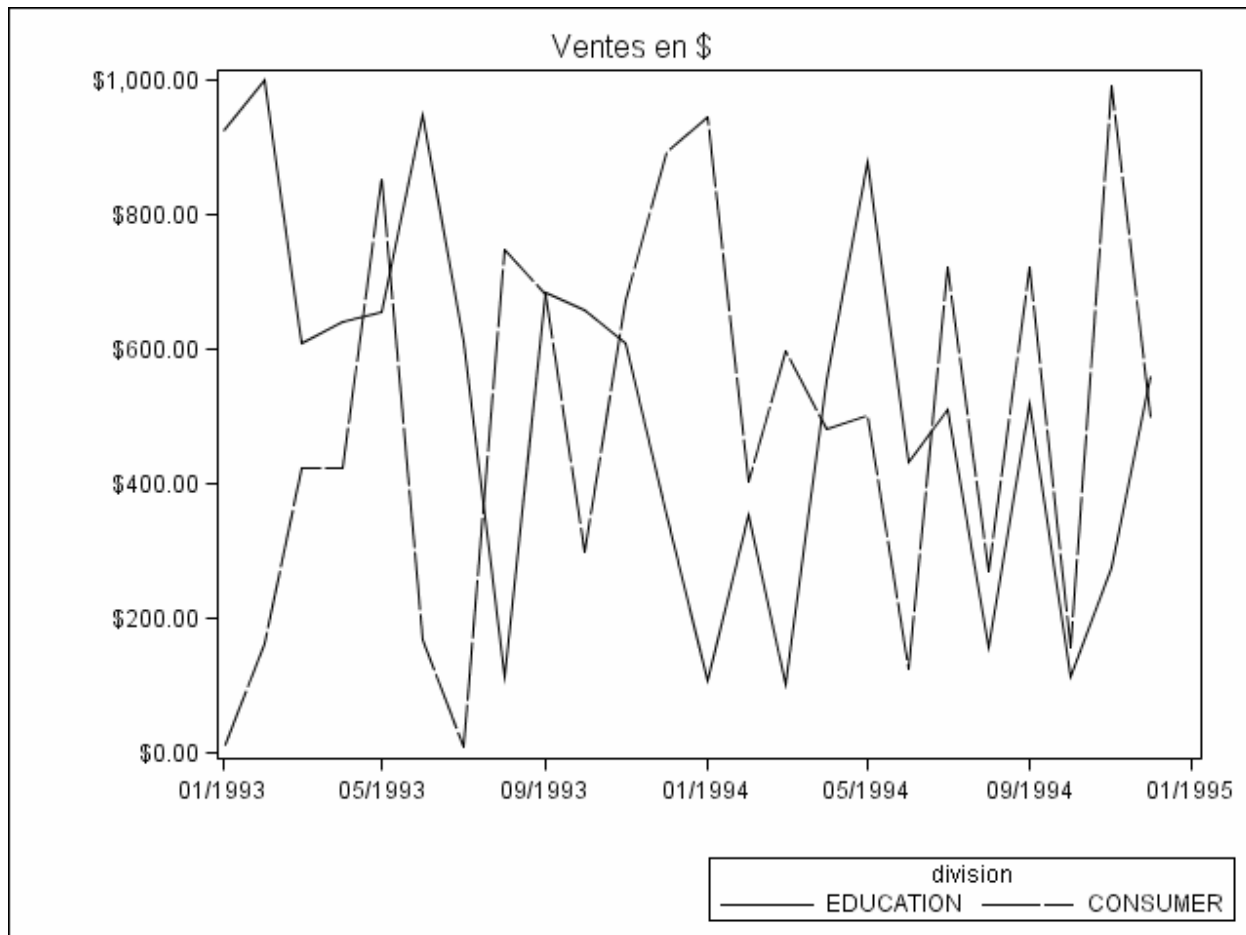
Si on ajoute ces instructions à l'intérieur d'un motif superposé (`LAYOUT OVERLAY`), ils apparaîtront à l'intérieur du graphique. Si on demande un autre motif (`GRIDDED` ou `LATTICE`), on peut les positionner hors du graphique.

```
ENTRY | ENTRYTITLE "texte" / VALIGN=TOP|CENTER|BOTTOM  
      | HALIGN=LEFT|CENTER|RIGHT  
      | PADTOP=marge PADBOTTOM=marge  
      | PADLEFT=marge PADRIGHT=marge ;
```

Outre les options de positionnement (VALIGN et HALIGN), on retrouve les options permettant de dimensionner des marges (options PAD...) identiques à celle de LAYOUT.

Exemple 10 – Ajouts de titres et de légendes à une courbe en GTL

```
PROC TEMPLATE ;
  DEFINE STATGRAPH exemples.courbe3 ;
  DYNAMIC varX varY varBy titre ;
  LAYOUT GRIDDED / COLUMNS=1 ;
    ENTRYTITLE titre / HALIGN=CENTER ;
    SERIESPLOT X=varX Y=varY / GROUP=varBy NAME="courbe"
              YAXISOPTS=(LABEL=" ") XAXISOPTS=(LABEL=" ") ;
    DISCRETELEGEND "courbe" / TITLE=varBy BORDER=TRUE
              HALIGN=RIGHT VALIGN=TOP;
  ENDLAYOUT ;
END ;
RUN ;
ODS HTML GPATH="c:\temp" STYLE=JOURNAL ;
DATA _NULL_ ;
  SET sashelp.prdsale (WHERE=(country="CANADA"
                            AND product="SOFA" AND region="EAST")) ;
  FILE PRINT ODS=(TEMPLATE="exemples.courbe3"
                 DYNAMIC=(varX="month" varY="actual" varBy="division"
                          titre="Ventes en $")) ;
  FORMAT month MMYYS7. ;
  PUT _ODS_ ;
RUN ;
ODS HTML CLOSE ;
```



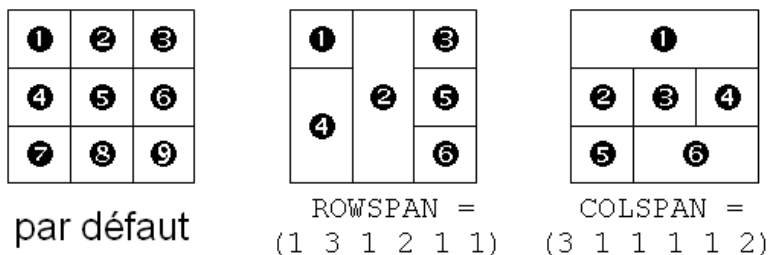
6. La fusion de cellules et l'imbrication de motifs

Pour créer des motifs complexes, deux solutions sont possibles : soit utiliser comme motif un unique LAYOUT GRIDDED en jouant sur les cellules fusionnées, soit encapsuler des motifs les uns dans les autres. Ces deux solutions peuvent cohabiter, comme le prouve l'exemple 11.

```
| LAYOUT GRIDDED / ROWSPAN=(fusions) | COLSPAN=(fusions) ;
```

Pour fusionner des cellules sur plusieurs lignes, on utilise l'option ROWSPAN. La figure 2 indique comment les valeurs de cette option associent à chaque cellule le nombre de lignes sur lesquelles elle sera fusionnée. De manière symétrique, l'option COLSPAN permet de fusionner sur plusieurs colonnes. Il est déconseillé d'utiliser simultanément ces deux options.

Figure 2 – Fusions de cellules dans le motif GRIDDED



Exemple 11 – Fusion de graphiques en GTL

```

PROC TEMPLATE ;
  DEFINE STATGRAPH exemples.bivar ;
  MVAR sysdate9 ;
  DYNAMIC var1 var2 ;
  LAYOUT GRIDDED / ROWS=4 COLUMNS=2 COLUMNSPAN=(1 1 2) ;
    LAYOUT OVERLAY / XAXISOPTS=(LABEL=var1) YAXISOPTS=(LABEL=" ") ;
      HISTOGRAM var1 ;
      ENTRY "En %" / VALIGN=TOP HALIGN=LEFT ;
    ENDLAYOUT ;
    LAYOUT OVERLAY / XAXISOPTS=(LABEL=var2) YAXISOPTS=(LABEL=" ") ;
      HISTOGRAM var2 ;
      ENTRY "En %" / VALIGN=TOP HALIGN=LEFT ;
    ENDLAYOUT ;
  LAYOUT GRIDDED / ROWS=2 COLUMNS=1 ;
    SCATTERPLOT X=var1 Y=var2 / XAXISOPTS=(LABEL=var1)
      YAXISOPTS=(LABEL=var2) ;
    ENTRYTITLE sysdate9 / HALIGN=RIGHT ;
  ENDLAYOUT ;
ENDLAYOUT ;
END ;
RUN ;
ODS HTML GPATH="c:\temp" ;
DATA _NULL_ ;
  SET livre.voitures (RENAME=(conso_auto=c poids=p)) ;
  FILE PRINT ODS=(TEMPLATE="exemples.bivar" DYNAMIC=(var1="p" var2="c")) ;
  PUT _ODS_ ;
RUN ;
ODS HTML CLOSE ;

```

