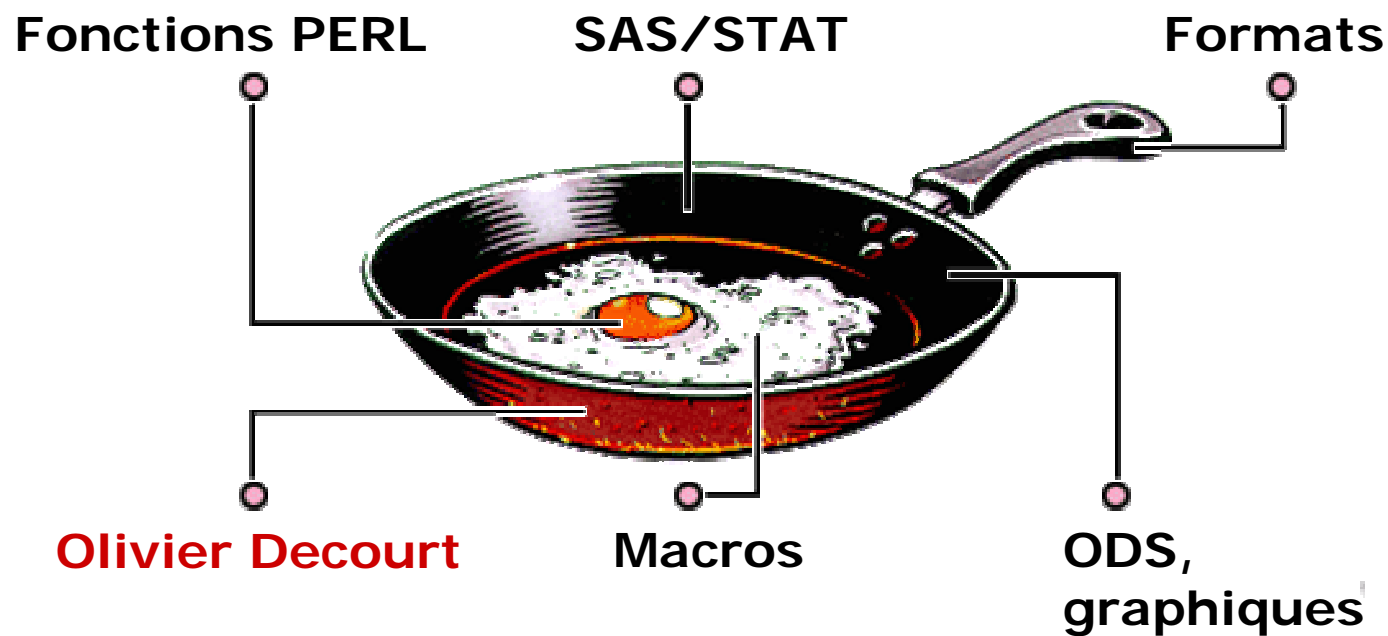


# Programmation : quoi de 9 ?

---





# Échanges avec Excel

- **PROC EXPORT vers Excel** (module ACCESS TO PC FILES) : possibilité de choisir la feuille du classeur Excel créée
- **LIBNAME vers MS Office** (module ACCESS TO PC FILES) : les feuilles d'un classeur Excel sont vues comme autant de tables différentes
- **Destination ODS TAGSETS.EXCELXP** (module SAS/BASE) : on écrit du XML compris par Excel 2003 et versions supérieures (ne concerne pas les productions graphiques) ; possibilité d'écrire dans plusieurs feuilles d'un même classeur

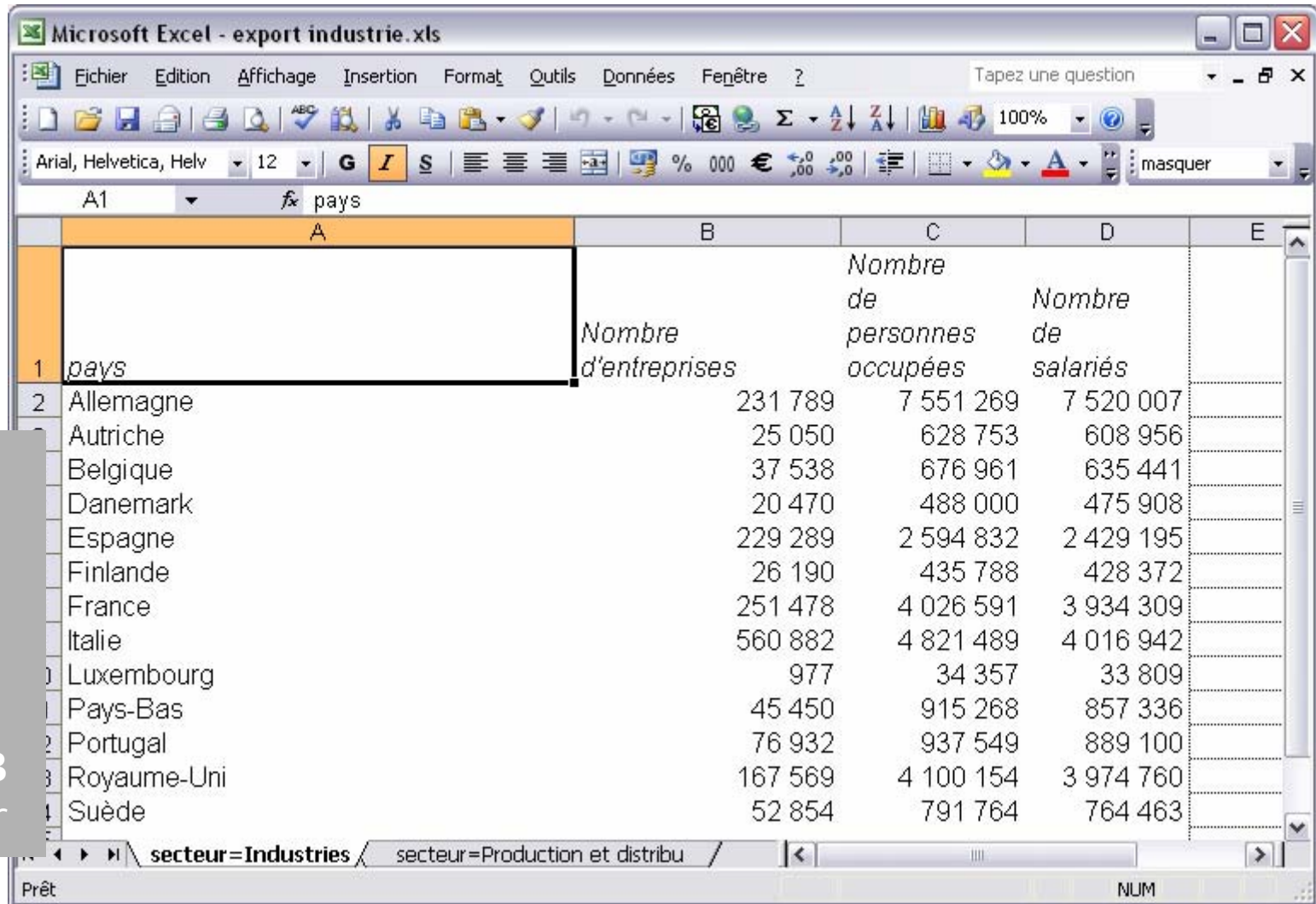


## Exemple de programme (1/2)

---

```
ODS TAGSETS.EXCELXP FILE = "d:\export industrie.xls"
                    STYLE = JOURNAL
OPTIONS (SHEET_INTERVAL = "BYGROUP"
        /* une feuille Excel par groupe BY */
        SUPPRESS_BYLINES = "YES") ;
TITLE1 "Données industrielles européennes" ;
FOOTNOTE1 "Année 2000 - Source : site Internet de l'Insee" ;
FOOTNOTE2 "Page %STR(&)p sur %STR(&)t" ;
/* Page 1 sur 3 mis à jour automatiquement */
PROC PRINT DATA = sas9.industrie LABEL NOOBS SPLIT = " " ;
    VAR pays coll col2 col3 / STYLE={TAGATTR="format:## ### ##0"} ;
    /* force le format de cellule Excel (ici, entiers avec
    séparateur de milliers */
    BY secteur ;
    FORMAT col: ; /* il est préférable d'éliminer les formats natifs
    de SAS et de gérer ceux des cellules Excel avec TAGATTR */
RUN ;
TITLE ;
FOOTNOTE ;
ODS TAGSETS.EXCELXP CLOSE ;
```

# Exemple de programme (2/2)



The screenshot shows a Microsoft Excel 2003 window titled "Microsoft Excel - export industrie.xls". The spreadsheet contains the following data:

<i>pays</i>	<i>Nombre d'entreprises</i>	<i>Nombre de personnes occupées</i>	<i>Nombre de salariés</i>
Allemagne	231 789	7 551 269	7 520 007
Autriche	25 050	628 753	608 956
Belgique	37 538	676 961	635 441
Danemark	20 470	488 000	475 908
Espagne	229 289	2 594 832	2 429 195
Finlande	26 190	435 788	428 372
France	251 478	4 026 591	3 934 309
Italie	560 882	4 821 489	4 016 942
Luxembourg	977	34 357	33 809
Pays-Bas	45 450	915 268	857 336
Portugal	76 932	937 549	889 100
Royaume-Uni	167 569	4 100 154	3 974 760
Suède	52 854	791 764	764 463



Ne fonctionne qu'avec Excel 2003 et supérieur

## Quelques petites facilités (1/4)

- **Nouveaux formats :**

OPTION LOCALE=FRENCH ;

→ Format **NLNUM.** affiche comme 999 999,99

- **Informat ANYDTDTE. :**

Permet de lire toutes les données dates

Si ambiguïté, OPTION DATESTYLE=DMY ; pour indiquer l'ordre le plus courant (jour / mois / année = DMY ; sinon, MDY, YMD, ...)



décembre 2006						
l	m	m	j	v	s	d
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

## Quelques petites facilités (2/4)

---

### ○ Nouvelles fonctions :

- **PRXMATCH**, **PRXCHANGE**, **PRXPOSN** pour permettre des opérations plus complexes que **INDEX**, **LIKE**, **TRANSLATE**, **TRANWRD** ou **SCAN** en utilisant des expressions régulières Perl (**PRX**)
- **IFN** et **IFC** renvoient des valeurs numériques ou caractère en fonction d'une condition et remplacent des **IF ... THEN ... ELSE**
- **STRIP** (= **LEFT**+**TRIM**), **FIRST** (= **SUBSTR** du 1<sup>er</sup> caractère), **CAT**, **CATS**, **CATT**, **CATX** pour concaténer des chaînes de caractères



## Quelques petites facilités (3/4)

---

### ○ Nouvelles fonctions :

- ANY... et NOT... (le texte contient au moins un type de caractère / ne contient aucun caractère de ce type), par exemple...

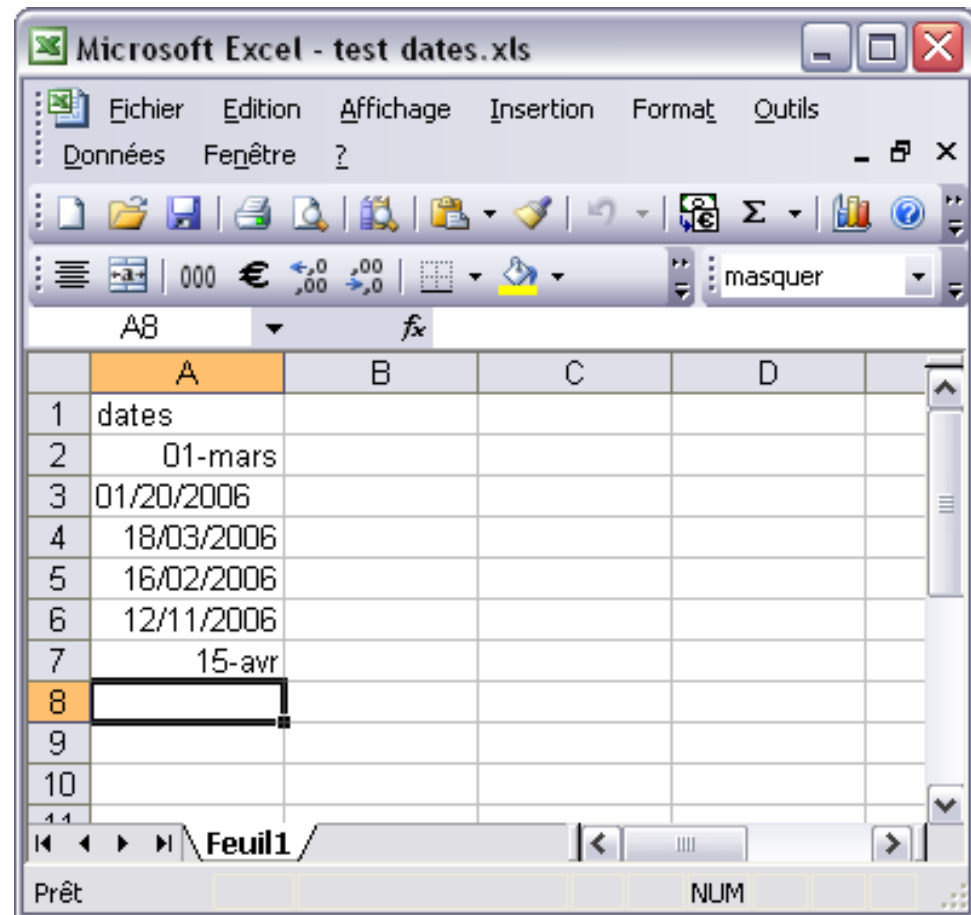
<b>ALNUM</b>	Lettres + chiffres
<b>ALPHA</b>	Lettres
<b>DIGIT</b>	Chiffres
<b>FIRST</b>	Caractères autorisés en 1er dans un nom SAS
<b>LOWER</b>	Minuscules
<b>NAME</b>	Caractères autorisés dans un nom SAS
<b>PUNCT</b>	Signes de ponctuation
<b>SPACE</b>	Espace / tabulation
<b>UPPER</b>	Majuscules



## Exemple de programme (1/5)

- Imaginons qu'on veuille importer une feuille Excel :

Seul problème potentiel : les dates sont stockées n'importe comment, et la colonne DATES contient des données comprises comme numériques à certaines lignes (quand les données sont justifiées à droite) et caractère à d'autres.



# Exemple de programme (2/5)

```
LIBNAME xl EXCEL "d:\demoSAS9\test dates.xls"  
MIXED = YES ;
```

The screenshot shows the SAS interface. On the left, the 'Explorateur' window displays 'Bibliothèques actives' (Active Libraries) with icons for Sas9, Maps, Sashelp, Sasuser, Work, and xl. Below it, another 'Explorateur' window shows the 'Contenu de xl' (Content of xl) with a 'Feuil1\$' (Sheet1) icon. On the right, the 'VIEWTABLE: xl.Feuil1\$' window displays a table with the following data:

	dates
1	01-mars
2	01/20/2006
3	18/03/2006
4	16/02/2006
5	12/11/2006
6	15-avr.

A yellow callout bubble contains the text: 'Avec l'option **MIXED=YES**, toutes les données sont importées dans une variable de type caractère'.

## Exemple de programme (3/5)

---

Objectif : conversion de toutes les dates

Etape 1 : création d'un informat pour les noms de mois en français

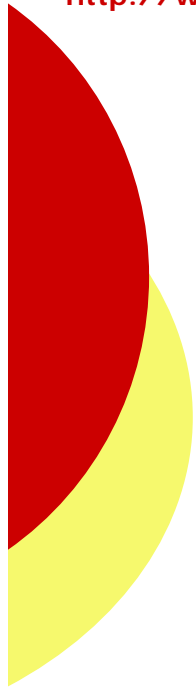
```
PROC FORMAT ;
  INVALUE $mois
    "janvier", "jan", "janv"      = 1
    "février", "fév", "févr"     = 2
    "mars", "mar"                 = 3
    "avril", "avr"                = 4
    "mai"                          = 5
    "juin"                         = 6
    "juillet", "jul", "juil"      = 7
    "août", "aou", "aoû"         = 8
    "septembre", "sep", "sept"    = 9
    "octobre", "oct", "octo"      = 10
    "novembre", "nov", "nove"     = 11
    "décembre", "dec", "déc"     = 12
  ;
RUN ;
```

## Exemple de programme (4/5)

---

Etape 2 : relecture des données et création d'une variable date

```
DATA work.dates (DROP = re mois) ;
  SET xl."feuille$"n ;
  dateSAS = INPUT(STRIP(dates),ANYDTDTE10.) ;
  IF dateSAS = . AND NOTDIGIT(COMPRESS(dates," /-")) THEN DO ;
    /* s'il y a autre chose que des chiffres, des espaces et
    des tirets dans la date */
    re = PRXPARSE("/^(\d\d).(\w+)/") ;
          /* 2 chiffres puis un mot */
  IF PRXMATCH(re,STRIP(dates)) THEN DO ;
    mois = PRXPOSN(re,2,STRIP(dates)) ; /* --> en nombre */
    dateSAS = MDY(INPUT(mois,$mois.),
                  SUBSTR(dates,1,2)+0,
                  YEAR(TODAY())) ;
  END ;
END ;
FORMAT dateSAS FRADFWDX. ;
RUN ;
```



## Exemple de programme (5/5)

---

dates	dateSAS
01 - mars	1er mars 2006
01/20/2006	20 janvier 2006
18/03/2006	18 mars 2006
16/02/2006	16 février 2006
12/11/2006	12 novembre 2006
15 - avr.	15 avril 2006

## Quelques petites facilités (4/4)

---

- **Macro-langage :**

- Les valeurs des macro-variables peuvent contenir jusqu'à 65 534 caractères.



- Routine **CALL SYMPUTX** avec plusieurs avantages :
  - conversion automatique de données numériques
  - 3<sup>e</sup> argument pour préciser la portée (locale/globale) de la macro-variable

# Exemple de programme (1/3)

---

```
%MACRO statDesc (table, variable) ;
  ODS EXCLUDE ALL ;
  ODS OUTPUT summary = work.stats ;
  PROC MEANS DATA = &table MEAN MEDIAN MIN MAX Q1 Q3 ;
    VAR &variable ;
  RUN ;
  ODS SELECT ALL ;
  DATA _NULL_ ;
    SET work.stats ;
    ARRAY stats _NUMERIC_ ;
    DO OVER stats ;
      CALL SYMPUTX(VNAME(stats),stats,"G") ;
    END ;
  RUN ;
%MEND statDesc ;
%statDesc (sas9.industrie, col10) ;
```

## Exemple de programme (2/3)

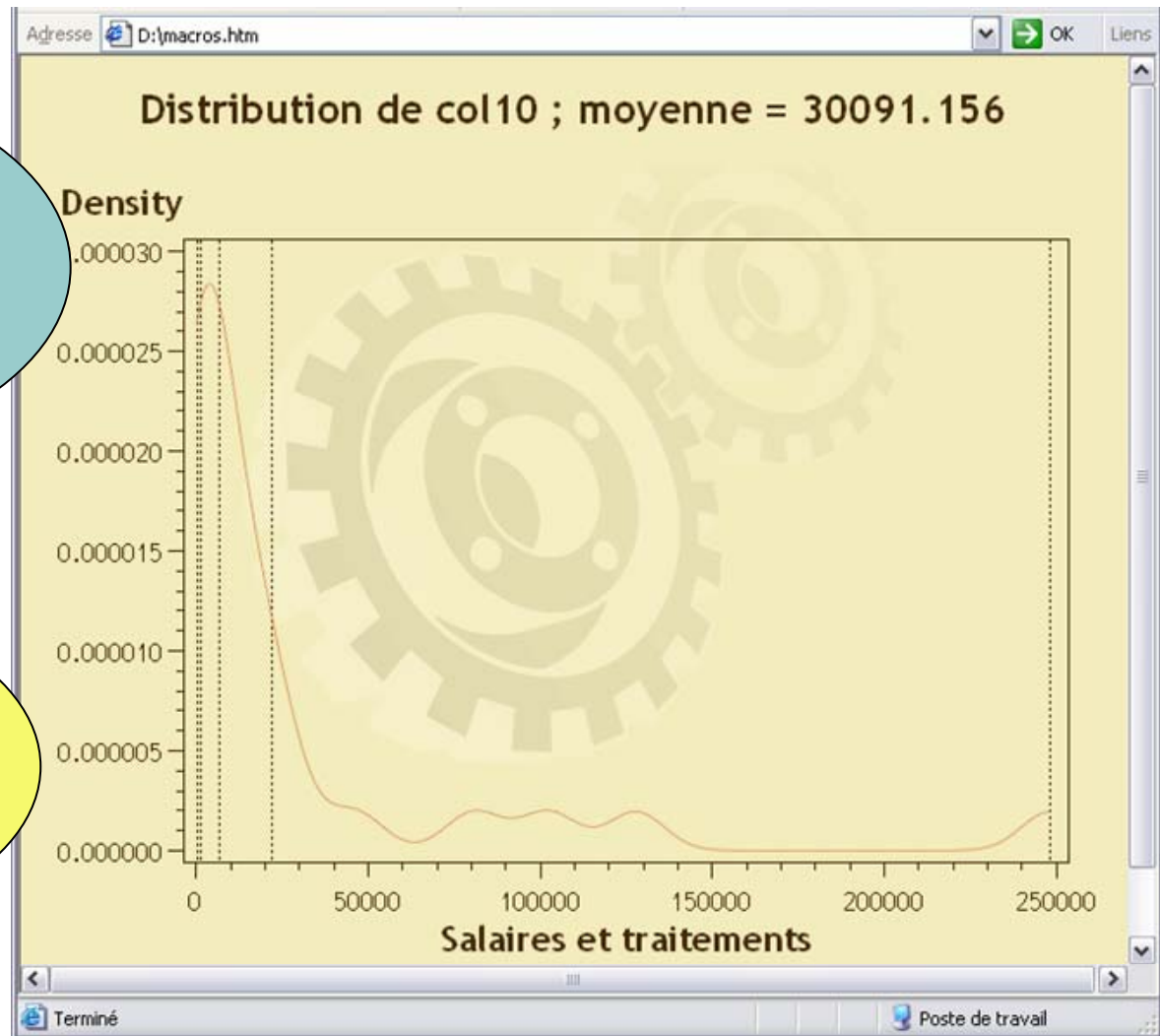
---

```
ODS EXCLUDE ALL ;
PROC KDE DATA = sas9.industrie ;
    UNIVAR col10 / OUT = work.col10 ;
RUN ;
ODS SELECT ALL ;
ODS HTML FILE = "d:\macros.htm" GPATH = "d:\" (URL = NONE)
    STYLE = GEARS ;
TITLE1 "Distribution de col10 ; moyenne = &col10_mean" ;
GOPTION DEVICE = ACTXIMG ;
SYMBOL i = join ;
PROC GPLOT DATA = work.col10 ;
    PLOT density * value
        / HREF = (&col10_min &col10_max &col10_
                median &col10_q1 &col10_q3)
        LHREF = 2 ;
    LABEL value = %SYSFUNC(VARLABEL(
                        %SYSFUNC(OPEN(sas9.industrie)),12)) ;
RUN ; QUIT ;
TITLE ;
ODS HTML CLOSE ;
```

# Exemple de programme (3/3)

Les valeurs des macro-variables ont permis de positionner des lignes de référence verticales et d'inclure la moyenne dans le titre.

Avec la GOPTION DEVICE=ACTXIMG, le style (ici, GEARS) est aussi appliqué au graphique.



# Des sorties sur mesure



- Dans l'étape Data, possibilité de construire des documents vers l'ODS sur mesure.
  - Objet **ODSOUT** à déclarer
  - Nombreuses méthodes à lui appliquer
  - Construction du résultat sur un principe de tableau (cf. HTML) avec possibilité de fusions de cellules en lignes et en colonnes
  - Plus compliqué que les procédures Print, Tabulate ou Report... mais aussi plus souple !



# Exemple de programme (1/4)

---

```
ODS LISTING CLOSE ;
ODS HTML FILE = "d:\objet ODS exemple1.htm" ;
TITLE1 "Données européennes 2000" ;
DATA _NULL_ ;
  SET sas9.industrie ;
  BY secteur NOTSORTED ;
  ARRAY colonnes col: ;   pctM = NMISS(OF col:)/DIM(colonnes) ;
  IF _N_ = 1 THEN DECLARE ODSOUT rapport() ;
  IF FIRST.secteur THEN DO ;
    rapport.TABLE_START() ; rapport.ROW_START() ;
    rapport.FORMAT_CELL(TEXT : secteur,
      OVERRIDES:"FONT_WEIGHT=BOLD BACKGROUND=PURPLE FOREGROUND=WHITE" ) ;
    rapport.FORMAT_CELL(TEXT : "% de valeurs manquantes",
      OVERRIDES:"FONT_WEIGHT=BOLD BACKGROUND=PURPLE FOREGROUND=WHITE" ) ;
    rapport.ROW_END() ;
  END ;
  rapport.ROW_START() ; rapport.FORMAT_CELL(TEXT : pays) ;
  SELECT ;
    WHEN(pctM > .20) rapport.FORMAT_CELL(TEXT : PUT(pctM*100,NLNUM12.2),
      OVERRIDES : "BACKGROUND=RED" ) ;
    WHEN(pctM > 0)   rapport.FORMAT_CELL(TEXT : PUT(pctM*100,NLNUM12.2),
      OVERRIDES : "BACKGROUND=ORANGE" ) ;
    OTHERWISE       rapport.FORMAT_CELL(TEXT : PUT(pctM*100,NLNUM12.2)) ;
  END ;
  IF LAST.secteur THEN DO ;
    rapport.TABLE_END() ;
  END ;
RUN ;
TITLE ;
ODS HTML CLOSE ;
ODS LISTING ;
```

# Exemple de programme (2/4)

On colorie les cases du tableau (ou des lignes entières) en fonction des valeurs

Données européennes 2000

Industries manufacturières	% de valeurs manquantes
Allemagne	0,00
Autriche	0,00
Belgique	0,00
Danemark	0,00
Espagne	0,00
Finlande	0,00
France	9,09
Italie	0,00
Luxembourg	9,09
Pays-Bas	0,00
Portugal	0,00
Royaume-Uni	0,00
Suède	0,00

Production et distribution d'électricité, de gaz et d'eau	% de valeurs manquantes
Allemagne	100,00

# Exemple de programme (3/4)

---

```
ODS LISTING CLOSE ;
ODS HTML FILE = "d:\objet ODS exemple2.htm" ;
TITLE1 "Données européennes 2000" ;
DATA _NULL_ ;
  SET sas9.industrie ;
  BY secteur NOTSORTED ;
  ARRAY colonnes col: ; pctM = NMISS(OF col:)/DIM(colonnes) ;
  IF _N_ = 1 THEN DECLARE ODSOUT rapport() ;
  IF FIRST.secteur THEN DO ;
    rapport.TABLE_START() ; rapport.ROW_START() ;
    rapport.FORMAT_CELL(TEXT : secteur, COLUMN_SPAN : 2,
      OVERRIDES : "FONT_WEIGHT=BOLD BACKGROUND=PURPLE FOREGROUND=WHITE") ;
    rapport.FORMAT_CELL(TEXT : "% de valeurs manquantes",
      OVERRIDES : "FONT_WEIGHT=BOLD BACKGROUND=PURPLE FOREGROUND=WHITE") ;
    rapport.ROW_END() ;
  END ;
  rapport.ROW_START() ; rapport.CELL_START() ;
  rapport.IMAGE(TEXT : CATS("images/",LOWCASE(pays),".jpg")) ; rapport.CELL_END() ;
  rapport.FORMAT_CELL(TEXT : pays) ;
  SELECT ;
    WHEN(pctM > .20) rapport.FORMAT_CELL(TEXT : PUT(pctM*100,NLNUM12.2),
      OVERRIDES : "BACKGROUND=RED") ;
    WHEN(pctM > 0)   rapport.FORMAT_CELL(TEXT : PUT(pctM*100,NLNUM12.2),
      OVERRIDES : "BACKGROUND=ORANGE") ;
    OTHERWISE       rapport.FORMAT_CELL(TEXT : PUT(pctM*100,NLNUM12.2)) ;
  END ;
  IF LAST.secteur THEN rapport.TABLE_END() ;
RUN ;
TITLE ;
ODS HTML CLOSE ;
ODS LISTING ;
```

# Exemple de programme (4/4)

On peut aisément inclure des images dans les tableaux générés

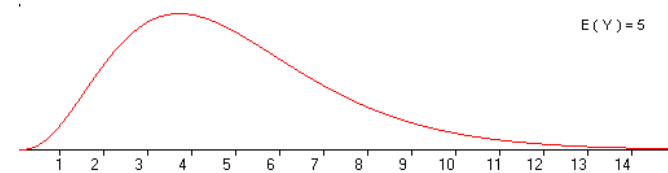
*Données européennes 2000*

Industries manufacturières		% de valeurs manquantes
	Allemagne	0,00
	Autriche	0,00
	Belgique	0,00
	Danemark	0,00
	Espagne	0,00
	Finlande	0,00
	France	9,09
	Italie	0,00

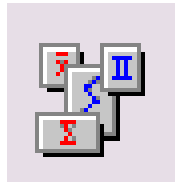
# Fonctions statistiques de l'étape DATA

---

- **Fonction RAND** pour remplacer RANUNI, RANNOR, RANBIN, etc.



- **Fonctions MEDIAN, IQR, PCTL, MAD** pour le calcul de statistiques par observation



- **Fonctions LARGEST et SMALLEST** pour récupérer la  $k^{\text{ème}}$  plus forte / faible valeur d'une série de variables

## Imputation de données manquantes

---

- La **proc MI** passe en production (MCMC)
- **Imputation de données qualitatives** expérimentale dans la **PROC MI**
- Nouvelle procédure **STDIZE** qui remplace la proc STANDARD (**imputation par moyenne, médiane, midrange**, éventuellement BY)



# Exemple de programme (1/4)

---

```
TITLE1 "Gestion des valeurs manquantes" ;
ODS LISTING CLOSE ;
PROC MI DATA = sas9.industrie
      (KEEP = col1 col22)
      OUT = work.mi
      NIMPUTE = 1
      MIN = 0 0 ;
VAR col1 col22 ;
RUN ;
```

## **PROC MI :**

Les options importantes sont  
NIMPUTE=1 (imputation  
simple), MIN et MAX pour  
border les valeurs imputées.

## Exemple de programme (2/3)

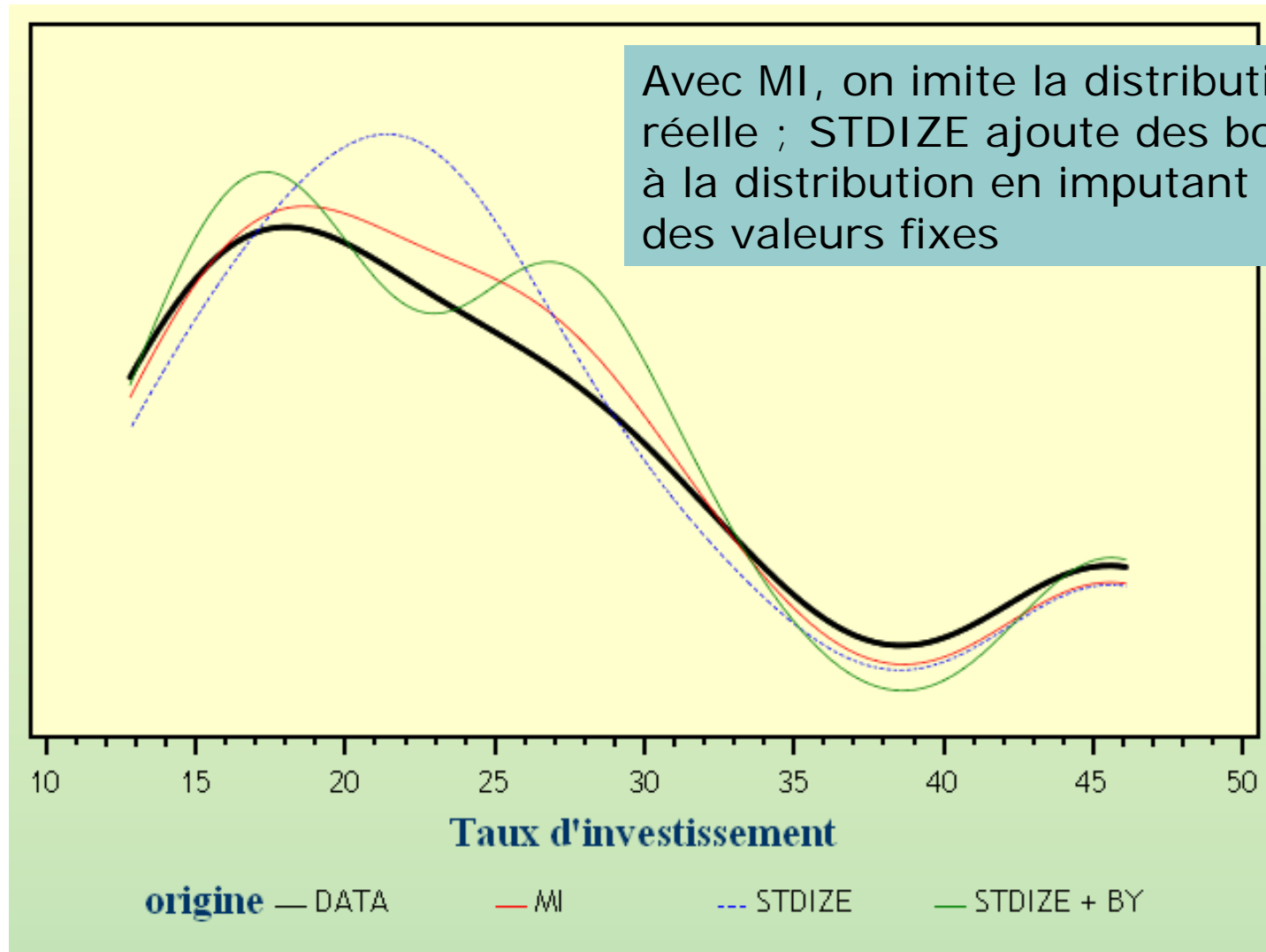
---

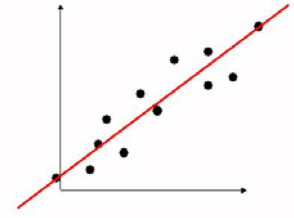
```
PROC STDIZE DATA = sas9.industrie
              (KEEP = secteur col1 col22)
              OUT = work.stdize
              REONLY METHOD = MEDIAN ;
BY secteur NOTSORTED ;
VAR col22 ;
RUN ;
PROC STDIZE DATA = sas9.industrie (KEEP = col22)
              OUT = work.stdize2
              REONLY METHOD = MEDIAN ;
VAR col22 ;
RUN ;
```

### **PROC STDIZE :**

Les options importantes sont REONLY (imputation) et METHOD pour choisir la statistique d'imputation (MEAN, MEDIAN, MIDRANGE).

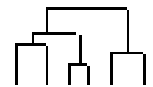
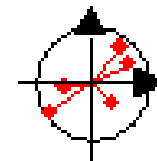
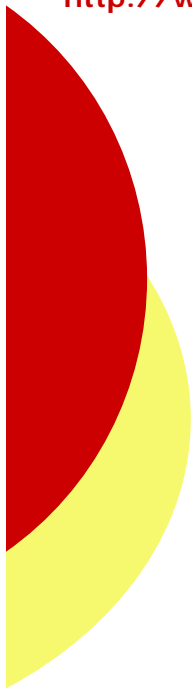
## Exemple de programme (3/3)





# Des graphiques statistiques

- **L'instruction ODS GRAPHICS**, conjuguée à l'ouverture d'un document PDF, RTF ou HTML, permet :
  - de produire automatiquement des graphiques pour certaines procédures (REG, ANOVA, PRINCOMP, ...)
  - d'utiliser de nouvelles options et instructions pour produire des graphiques dans une trentaine de procédures statistiques (CORR, LOGISTIC, ARIMA, ...)

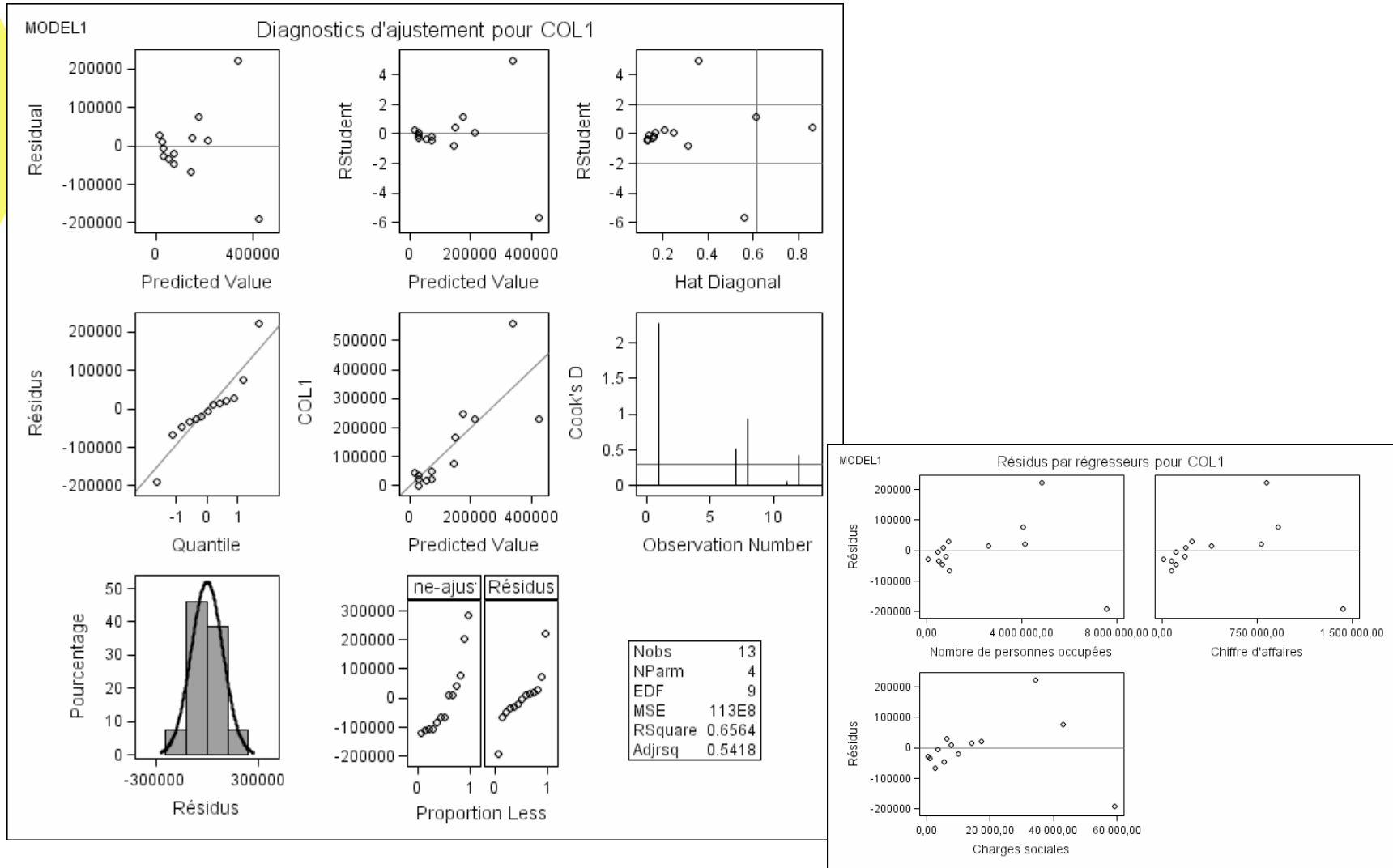


## Exemple de programme (1/3)

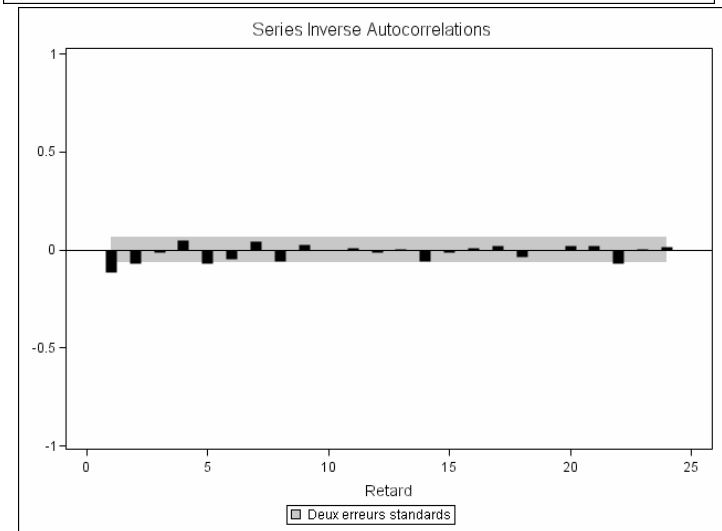
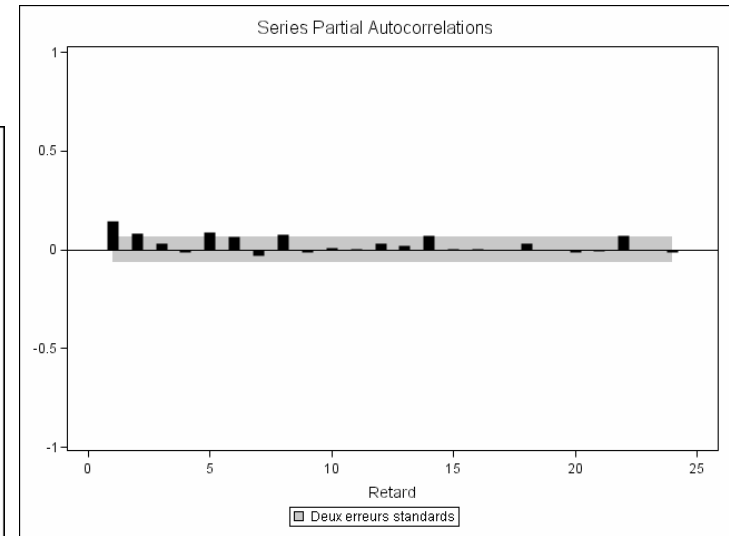
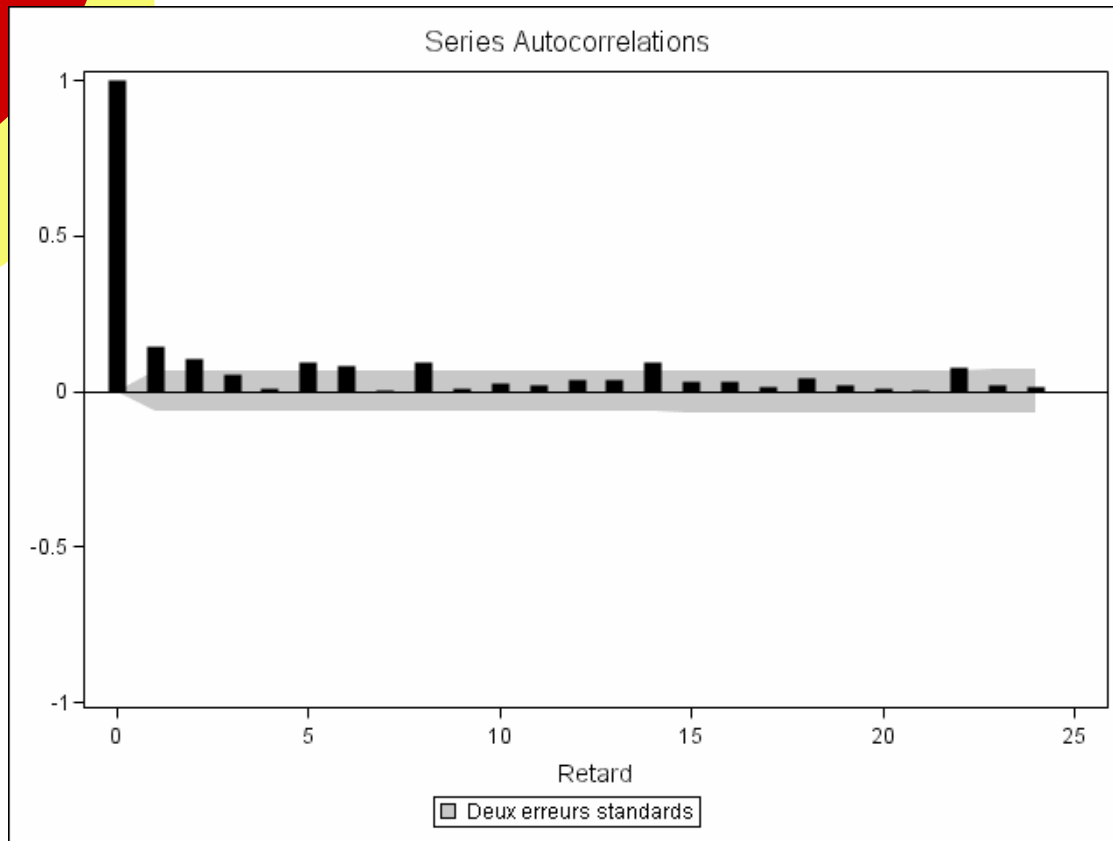
---

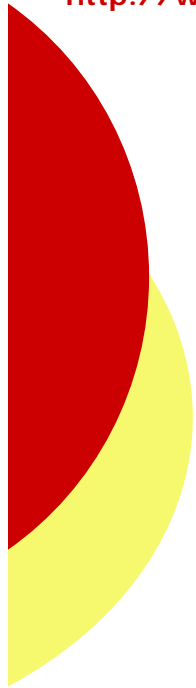
```
ODS HTML FILE = "d:\ods graphics1.htm"
      GPATH = "d:\" (URL = NONE)
      NEWFILE = PROC
      STYLE = JOURNAL ;
ODS GRAPHICS ON / IMAGEFMT = STATICMAP
;
/* l'option STATICMAP permet d'activer
des info-bulles */
PROC REG DATA = sas9.industrie ;
      WHERE      secteur      =      "Industries
manufacturières" ;
      MODEL col1 = col2 col7 col11 / R ;
```

# Exemple de programme (2/3)



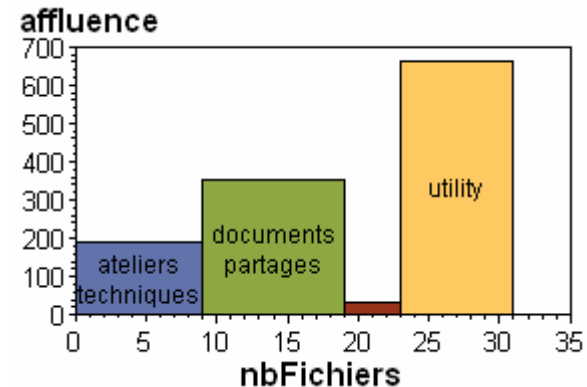
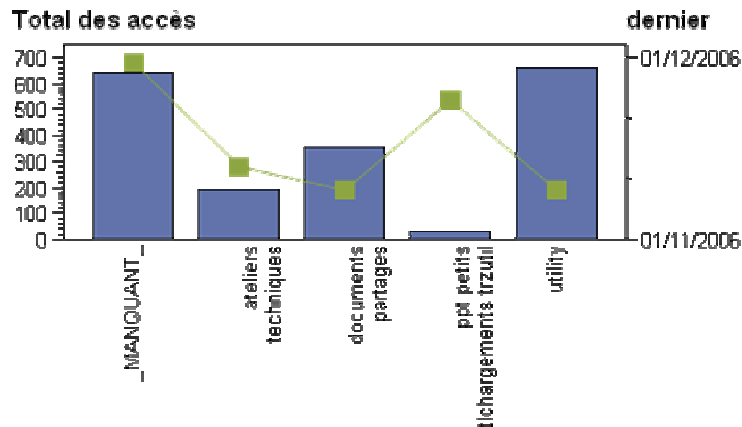
# Exemple de programme (3/3)





# Nouvelles procédures graphiques

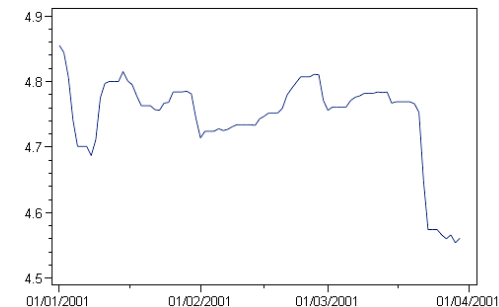
- La **proc GBARLINE** permet de superposer un histogramme et une courbe sans passer par un Annotate
- La **proc GAREABAR** permet de représenter deux variables continues en plus d'une variable catégorielle, en jouant sur la surface de bâtons horizontaux ou verticaux



# Un traitement nouveau des séries temporelles

---

- **Forecast Studio** : dix procédures dont, principalement, HPF et HPFDIAGNOSE
- **HPFDIAGNOSE** : sélection automatique de modèles dans plusieurs familles (dont ARIMA, SARIMA, ARIMAX, lissage exponentiel) et détection d'évènements « level-shift »
- **HPF** : prévision automatique de séries par lissage, transformations de Box-Cox sur les données à prédire



## Exemples de programmes (1/9)

---

```
PROC          HPFDIAGNOSE          DATA          =
sas9 .banques

                PRINT = ALL
                CRITERION = AIC ;
WHERE dates <= "01oct2001"d ;
ID dates INTERVAL=DAY ;
FORECAST euribor_3m ;
INPUT euribor_6m ;
ARIMAX ;
AUTOMODEL ;

RUN ;
```

# Exemples de programmes (2/9)

Dickey-Fuller Unit Root Test				
Type	Rho	Pr < Rho	Tau	Pr < Tau
Zero Mean	0.05	0.6956	0.19	0.7407
Single Mean	-1.27	0.8598	-0.91	0.7842
Trend	1.42	0.9995	0.57	0.9995

Seasonal Dickey-Fuller Unit Root Test(Seasonality=7)				
Type	Rho	Pr < Rho	Tau	Pr < Tau
Zero Mean	-1.62	0.4131	-0.75	0.2883
Single Mean	-53.16	0.0011	-5.32	0.0001

ARIMA Model Specification												
Variable	Functional Transform	Constant	p	d	q	P	D	Q	Seasonality	Model Criterion	Statistic	Status
Euribor_3M	NONE	NO	2	1	0	0	0	0	7	AIC	-7330.95	OK

ARIMA Input Selection								
Input Variable	Selected	Functional Transform	d	D	Delay	Numerator	Denominator	Status
Euribor_6M	YES	NONE	1	0	0	2	0	OK

# Exemples de programmes (3/9)

ARIMA Outlier Selection					
Variable	Type	Obs	Time	Chi-Square	Approx Pr > ChiSq
Euribor_3M	LS	272	29SEP1999	14123.8	<.0001
	LS	992	18SEP2001	1556.27	<.0001

Exponential Smoothing Model Specification					
Variable	Functional Transform	Selected Model	Component	Model Criterion	Statistic
Euribor_3M	NONE	DAMPTREND	LEVEL	AIC	-7341.54
			TREND		
			DAMP		

Best Model Specification															
Variable	Functional Transform	Constant	p	d	q	P	D	Q	Seasonality	Outlier	Input	Model Criterion	Statistic	Status	
Euribor_3M	NONE	NO	2	1	0	0	0	0	7	2	1	AIC	-9566.58	OK	

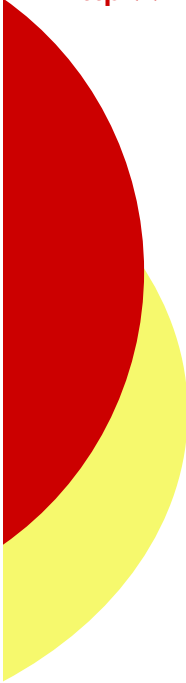
## Exemples de programmes (4/9)

---

```
ODS GRAPHICS ON ;
PROC HPF DATA = sas9.banques
    LEAD = 30
        OUT = work.prev
    PRINT = ALL
    PLOT = ALL ;
WHERE dates <= "01oct2001"d ;
ID dates INTERVAL=DAY ;
FORECAST euribor_3m / MODEL =
BESTALL SELECT = AIC ;

RUN ;
```

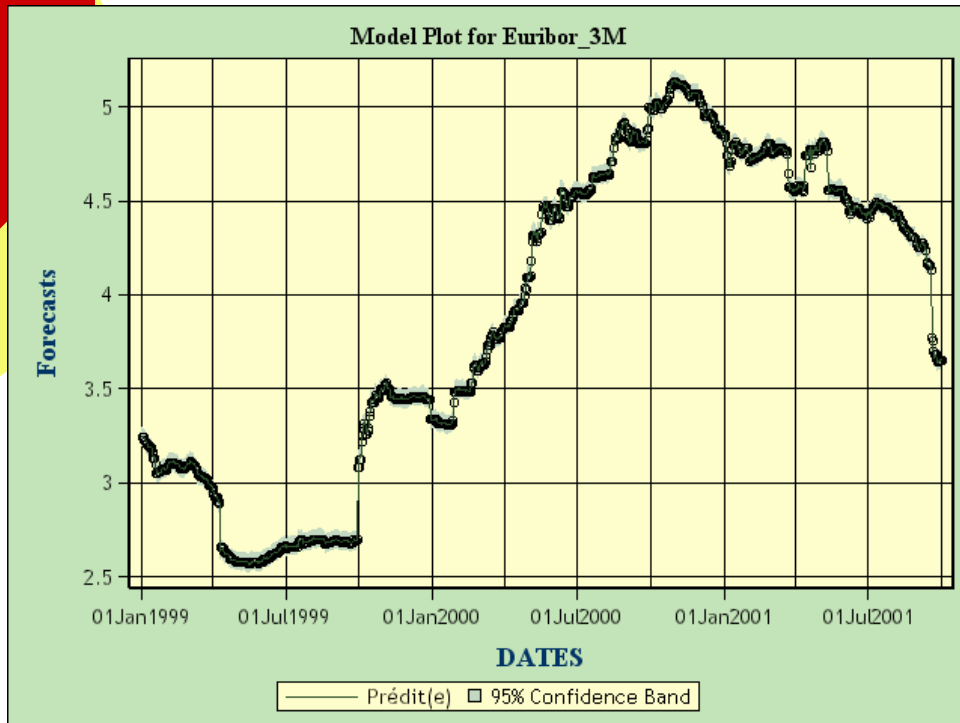
# Exemples de programmes (5/9)



Descriptive Statistics	
Variable	Euribor_3M
Number of Observations	1005
Number of Missing Observations	0
Minimum	2.57
Maximum	5.14
Mean	3.911761
Standard Deviation	0.837523

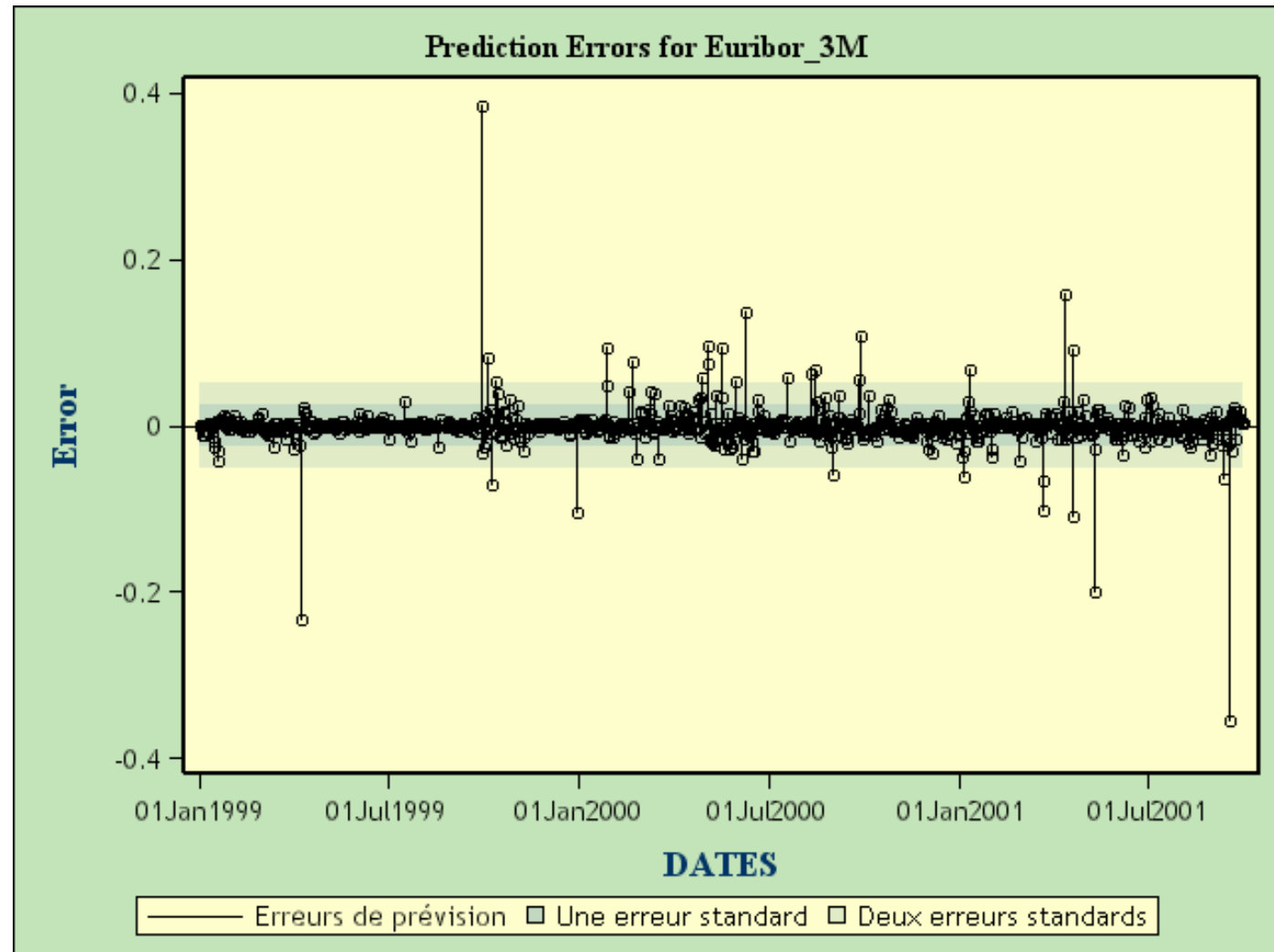
Model Selection Criterion = AIC		
Model	Statistic	Selected
Simple Exponential Smoothing	-7312.9664	
Double Exponential Smoothing	-7195.0344	
Linear Exponential Smoothing	-7333.2260	
Damped-Trend Exponential Smoothing	-7341.5438	Yes

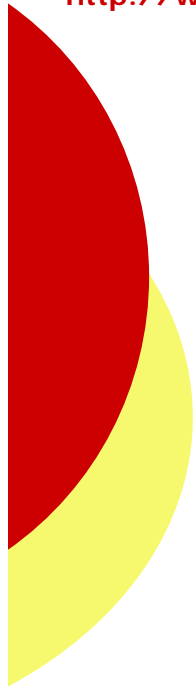
# Exemples de programmes (6/9)



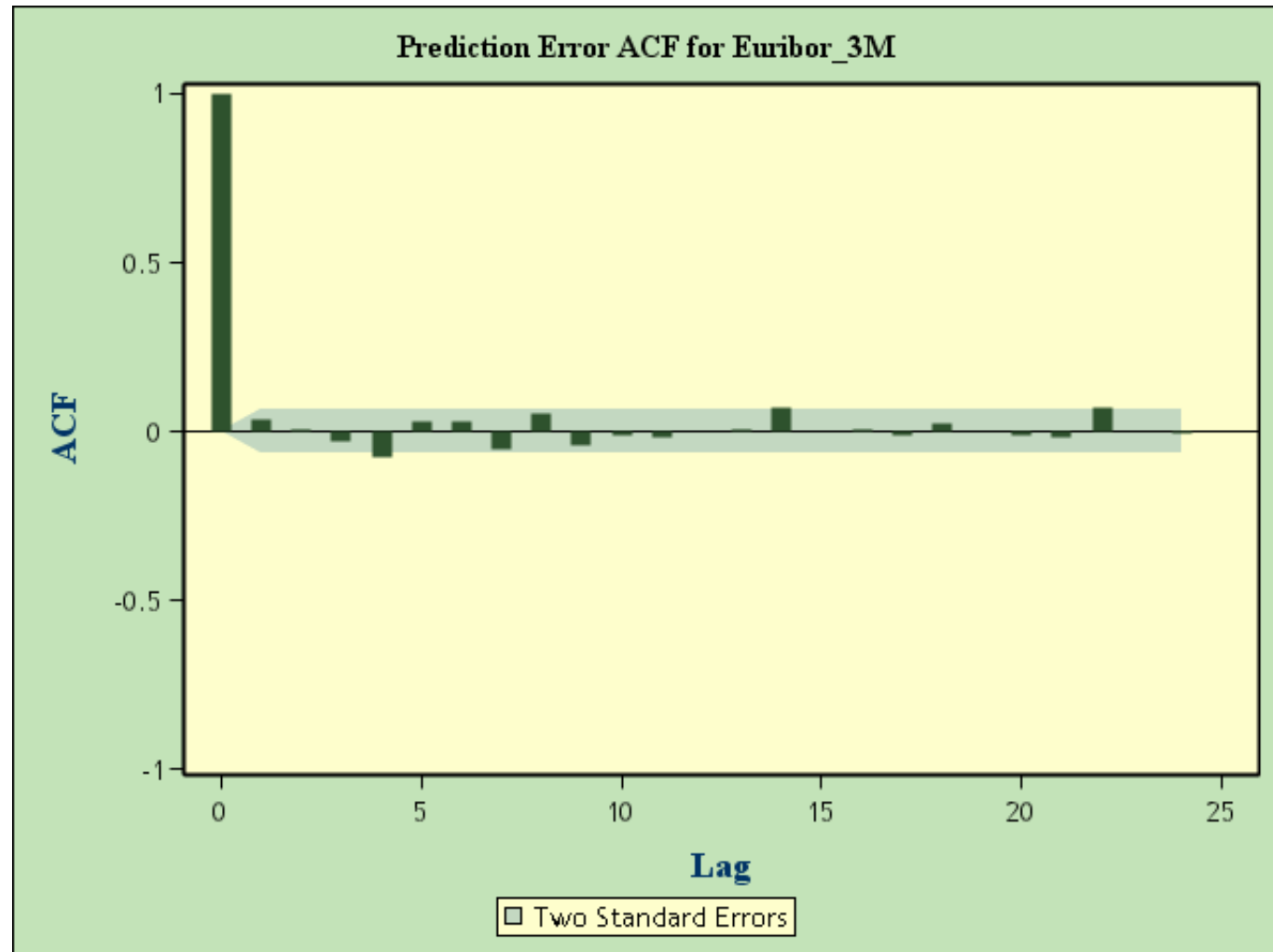
Damped-Trend Exponential Smoothing Parameter Estimates				
Parameter	Estimation	Standard Error	Valeur du test t	Approx Pr >  t
Level Weight	0.99900	0.02905	34.38	<.0001
Trend Weight	0.09853	0.02939	3.35	0.0008
Damping Weight	0.87905	0.04325	20.33	<.0001

# Exemples de programmes (7/9)





# Exemples de programmes (8/9)



# Exemples de programmes (9/9)

