

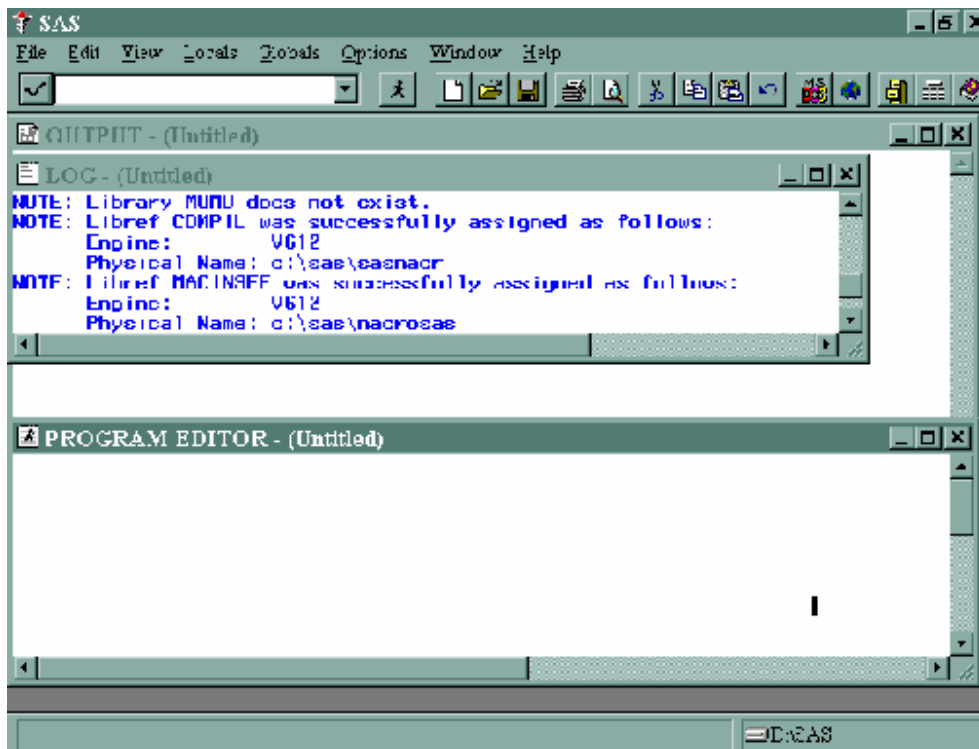
## Introduction au logiciel SAS. Support de cours.

Olivier Godechot

### Préambule. A quoi ressemble la base de données?

Pour toute utilisation: questionnaire + dictionnaire des codes. Connaître les noms dans la base de données des variables. Connaître le codage des items: en général, c'est un codage avec des chiffres qui est adopté (gain de places).

### Présentation du logiciel.



Graphique n°1:présentation des fenêtres de SAS.

### Les trois fenêtres d'importance.

*Program Editor*: Écriture du programme.

*LOG*: visualisation du déroulement du programme et des erreurs

*Output* : visualisation des résultats du programme (quand il n'y a pas d'erreurs).

Sous chacune de ces fenêtres des menus: *File Edit View Locals Globals Help*.

→ Regardons les menus déroulants de la fenêtre Program Editor.

*File* (**OPEN** ou **INFILE** ouverture de fichier, enregistrement de fichiers, impression...);

*Edit* (*chercher, remplacer, sélectionner, désélectionner, annuler...*);

*Locals* (**Submit** soumettre le programme, **Recall text** : Rappeler le texte,...);

*Globals* (changement de fenêtre);

*Help* (aide en ligne laconique).

## L'écriture de programmes.

- Aperçu sur la grammaire générale de SAS: Fonction, option, instruction, procédure. Pour écrire un programme, il est utile de connaître quelques rudiments sur la grammaire générale du langage SAS, ne serait-ce que pour se retrouver dans le manuel maîtriser SAS. A regarder l'index, l'on remarque que certains termes identiques peuvent avoir plusieurs renvois. Par exemple Keep (garder) peut être désigné comme une instruction ou une option.

**Procédure:** il s'agit d'un programme tout fait comme faire des tableaux croisés, une régression, calculer des moyennes, etc.. Une procédure se déclare toujours par le terme "PROC". Elle se termine par l'instruction de fin de procédure "run;"

**Fonction:**  $a=f(b)$ , une nouvelle variable a est fonction d'une autre variable b, il existe toutes sortes de fonctions mathématiques, arithmétiques, mais aussi textuelles.

**Instruction:** C'est assez difficile à définir. Une suite d'instruction constitue un programme. La différence avec la procédure est la suivante. **Toute instruction se termine par un point virgule!!!!**

Exemple: `Options nocenter pagesize=100;`

**Option:** précision optionnelle en marge d'une instruction ou d'une procédure ; elle ne se termine pas forcément par un point virgule, écriture entre parenthèses et avec un signe d'égalité fréquente.

## Étape pour construire un programme.

1) Libname: nom d'un chemin pour que SAS trouve la base de donnée.

2) Étape Data: modification et ou construction de la base de données.

3) Étape Proc: procédures statistiques sur la nouvelle base de données ainsi construite.

Lancement du programme.

Le programme a-t-il réussi? Vérification de la fenêtre Log.

Analyse des résultats. Impression, enregistrement du programme ou modification.

Libname:

Les bases de données SAS se présentent comme des fichiers de type **pif.sd2** sous windows.

*Principe numéro 1: Sas est très bête, il faut tout lui dire.*

*Il ne connaît pas les répertoires. On va donc lui indiquer un chemin. Dans l'exemple "toto" sera le nom de répertoire pour le logiciel sas.*

*pour PC*

```
libname toto 'c:\amoi\base\';
```

S'il existe un fichier pif.sd2 dans le répertoire c:\amoi\base\, alors, pour SAS, il s'appellera toto.pif. (Cette opération n'est à faire qu'une seule fois par session SAS).

Remarque: il est possible d'effectuer directement des procédures en appelant la base "toto.pif" sans faire d'étape Data.

```
proc freq data=toto.pif;
tables sexe*Jungl;
run;
```

### Résultat

```
TABLE OF SEXE BY JUNGL
SEXE      JUNGL
Frequency|
Percent  |
Row Pct  |
Col Pct  |1          |2          |9          | Total
-----+-----+-----+-----+
1         |    241 |    1260 |     2   |  1503
         |    8.10|    42.34|    0.07 |  50.50
         |   16.03|    83.83|    0.13 |
         |   56.57|    49.47|   66.67 |
-----+-----+-----+-----+
2         |    185 |    1287 |     1   |  1473
         |    6.22|    43.25|    0.03 |  49.50
         |   12.56|    87.37|    0.07 |
         |   43.43|    50.53|   33.33 |
-----+-----+-----+-----+
Total      426      2547      3      2976
          14.31   85.58   0.10  100.00
Frequency Missing = 2175
```

*A ce stade, il est d'ailleurs très utile de faire un proc format (voir supra), pour connaître les noms et les caractéristiques des variables (caractères ou numériques).*

*Mais le problème c'est qu'il faut généralement modifier la base de données avant de lancer des procédures. On distingue ainsi une étape data et une étape proc. L'étape data est l'étape où l'on modifie la base de données, et l'étape proc celle où l'on modifie les procédures.*

## Étape Data

*NB : Il existe de nombreuses manières d'écrire des programmes qui conduisent au même résultat.*

*On commence par créer une table temporaire dans laquelle on met une table permanente.*

```
Data A; set toto.pif;
```

*Ceci veut dire, je crée une base temporaire (qui dure le temps de la session SAS), et j'y mets (Set) la base toto.pif.*

*Avec les instructions keep ou drop, on peut limiter le nombre de variables (ce qui permet d'aller plus vite!!!).*

```
Data A (keep=sexe jungl age); set toto.pif;
```

*ou bien*

```
Data A (drop=sexe salaire obsta faspecl); set toto.pif;
```

*et surtout on peut modifier les variables et en créer de nouvelles. Par exemple, je veux calculer le temps de trajet.*

```
libname toto 'c:\amoi\base\';

Data a; set toto.pif;
if quest=1 then _TRAJH = ATRAJH ;
else if quest=2 then _TRAJH= TTRAJH ;
if quest=1 then _TRAJM = ATRAJM ;
else if quest=2 then _TRAJM= TTRAJM ;
trajet=_trajh+_trajm/60;
if _trajh=9 then trajet=_trajm/60;
if _trajm=99 then trajet=_trajh;
if (_trajh=8 and _trajm=88) or (_trajh=9 and _trajm=99) or
_trajh="" or _trajm="" then trajet=0;
trajet9=((_trajh=8 and _trajm=88) or (_trajh=9 and
_trajm=99) or _trajh="" or _trajm="" );
trajet2=trajet*trajet;
label trajet='Tps/HTrajet';
label trajet9='TrajetAtyp';
Homme=(sexe=1);
run;
```

*L'étape Data se termine toujours par run.*

```
run;
```

*L'ensemble des modifications de la base toto.pif, écrites entre “ set toto.pif; ” et “ run ; ” sont prises en compte.*

Opérateurs “ booléens ” classiques if ... then, else, and, or, do ... end, NE, <>, etc...

Comment restreindre la taille d'une base de données ? Avec “ if ” pour réduire le nombre d'observations et “ keep ” pour les variables.

```
If sexe="1";
```

On ne prend que les hommes.

```
Keep sexe quest;
```

On garde la variable quest et la variable sexe.

## **Variables numériques et variables caractères.**

*La plupart des variables sont au format caractère même s'il n'y a que des chiffres (car cela permet de tenir moins de place). Or certaines procédures demandent exclusivement des variables numériques. Voici la technique la plus simple de transformation: nouveau\_nom= ancien\_nom\*1.*

```
conge=anbcong*1;
```

*Il est conseillé de prendre l'habitude d'écrire les valeurs des variables caractères entre guillemets, même si ce sont des nombres :*

```
if sexe="2" then msex="femme";  
else if sexe="1" then msex="homme";
```

### **Les non-réponses.**

Les non-réponses pour SAS.

Si la variable est de type numérique, elles sont codées par un point.

Si la variable est de type caractère, noté par un blanc ' '

Elles ne sont pas prises en compte dans les calculs.

Les non-réponses pour l'INSEE.

Dans les bases INSEE, normalement, les non-questions sont notés par une non-réponse SAS et les non réponses par '9' ou '99' ou '999', etc..

**Pour les moyennes, il est impératif de virer les non réponses codées en 9 (le nombre de 9 dépend de la largeur de la variable).**

```
conge=anbcong*1;  
if conge=99 then conge=.;
```

### **Enregistrer un fichier**

```
libname tata 'c:\amoi\base';  
data tata.zaza; set a;
```

**Je trouve dans mon répertoire c:\amoi\base\zaza.sd2** qui est une copie permanente du fichier temporaire a.

Ne pas le faire pour des gros fichiers (place disque !!!).

### **Proc Contents**

```
Proc contents data=toto.argent2;  
run;
```

#### **Résultats**

CONTENTS PROCEDURE

Data Set Name:	TOTO.ARGENT2	Observations:	5151
Member Type:	DATA	Variables:	173
Engine:	V610	Indexes:	0
Created:	19:29 Tuesday, November 24, 1998	Observation Length:	416
Last Modified:	19:30 Tuesday, November 24, 1998	Deleted Observations:	0
Protection:		Compressed:	NO
Data Set Type:		Sorted:	NO
Label:			

-----Engine/Host Dependent Information-----

Data Set Page Size: 12800  
Number of Data Set Pages: 174  
File Format: 607  
First Data Page: 2  
Max Obs per Page: 30  
Obs in First Data Page: 8

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Label
38	AEVOLU	Char	1	86	Evolution depuis 5 ans
16	AGE	Num	4	39	Age Kish Num
17	AGE1	Char	1	43	Age 8 mod
26	AGEPR	Num	8	58	Age personne reference
3	AGEVCJ	Num	8	11	Age conjoint

...

## Proc Print

```
proc print data=a (OBS=10);var trajet trajet9 quest; run;
```

OBS	TRAJET	TRAJET9	QUEST
1	0.33333	0	1
2	0.50000	0	1
3	0.00000	1	3
4	0.50000	0	1
5	0.00000	1	3
6	0.66667	0	1
7	0.00000	1	1
8	2.00000	0	1
9	1.16667	0	1
10	2.25000	0	1

## Proc Freq

Programme :

```
data b (keep=sexe viecou acommod tcommod sexecou _commod);  
set toto.pif;  
if quest=2 then _COMMOD= TCOMMOD ;  
if quest=1 then _COMMOD= ACOMMOD ;  
if _commod=9 then _commod="";  
sexecou=10*sexe+viecou;  
run;
```

```
proc freq data=b; tables (sexe viecou sexecou)*_commod  
sexecou/cellchi2 chisq;  
run;
```

Résultats :

```
TABLE OF SEXE BY _COMMOD  
  
SEXE          _COMMOD  
Frequency    |  
Cell Chi-Square|  
Percent      |  
Row Pct      |  
Col Pct      |1      |2      | Total
```

1	1203	294	1497
	0.0095	0.0384	
	40.57	9.92	50.49
	80.36	19.64	
	50.63	49.92	
2	1173	295	1468
	0.0097	0.0392	
	39.56	9.95	49.51
	79.90	20.10	
	49.37	50.08	
Total	2376	589	2965
	80.13	19.87	100.00

Frequency Missing = 2186

STATISTICS FOR TABLE OF SEXE BY \_COMMOD

Statistic	DF	Value	Prob
Chi-Square	1	0.097	0.756
Likelihood Ratio Chi-Square	1	0.097	0.756
Continuity Adj. Chi-Square	1	0.070	0.791
Mantel-Haenszel Chi-Square	1	0.097	0.756
Fisher's Exact Test (Left)			0.640
(Right)			0.395
(2-Tail)			0.782
Phi Coefficient		0.006	
Contingency Coefficient		0.006	
Cramer's V		0.006	

Effective Sample Size = 2965

Frequency Missing = 2186

WARNING: 42% of the data are missing.

TABLE OF VIECOU BY \_COMMOD

VIECOU	_COMMOD		Total
Frequency	1	2	
Cell Chi-Square			
Percent			
Row Pct			
Col Pct			
1	1632	419	2051
	0.0894	0.3611	
	55.06	14.14	69.20
	79.57	20.43	
	68.69	71.26	
2	744	169	913
	0.2008	0.8112	
	25.10	5.70	30.80
	81.49	18.51	
	31.31	28.74	
Total	2376	588	2964
	80.16	19.84	100.00

Frequency Missing = 2187

STATISTICS FOR TABLE OF VIECOU BY \_COMMOD

Statistic	DF	Value	Prob
Chi-Square	1	1.462	0.227
Likelihood Ratio Chi-Square	1	1.477	0.224
Continuity Adj. Chi-Square	1	1.344	0.246
Mantel-Haenszel Chi-Square	1	1.462	0.227
Fisher's Exact Test (Left)			0.123
(Right)			0.896
(2-Tail)			0.232

Phi Coefficient -0.022  
 Contingency Coefficient 0.022  
 Cramer's V -0.022

Effective Sample Size = 2964  
 Frequency Missing = 2187  
 WARNING: 42% of the data are missing.

TABLE OF SEXECOU BY \_COMMOD

SEXECOU	_COMMOD		Total
Frequency	1	2	
11	864	228	1092
Cell Chi-Square	0.1476	0.5966	
Percent	29.15	7.69	36.84
Row Pct	79.12	20.88	
Col Pct	36.36	38.78	
12	339	65	404
Cell Chi-Square	0.7083	2.8622	
Percent	11.44	2.19	13.63
Row Pct	83.91	16.09	
Col Pct	14.27	11.05	
21	768	191	959
Cell Chi-Square	0.0007	0.003	
Percent	25.91	6.44	32.35
Row Pct	80.08	19.92	
Col Pct	32.32	32.48	
22	405	104	509
Cell Chi-Square	0.0224	0.0906	
Percent	13.66	3.51	17.17
Row Pct	79.57	20.43	
Col Pct	17.05	17.69	
Total	2376	588	2964
	80.16	19.84	100.00

Frequency Missing = 2187

STATISTICS FOR TABLE OF SEXECOU BY \_COMMOD

Statistic	DF	Value	Prob
Chi-Square	3	4.431	0.218
Likelihood Ratio Chi-Square	3	4.610	0.203
Mantel-Haenszel Chi-Square	1	0.055	0.815
Phi Coefficient		0.039	
Contingency Coefficient		0.039	
Cramer's V		0.039	

Effective Sample Size = 2964  
 Frequency Missing = 2187  
 WARNING: 42% of the data are missing.

SEXECOU	Frequency	Percent	Cumulative Frequency	Cumulative Percent
11	1689	32.8	1689	32.8
12	665	12.9	2354	45.7
21	1782	34.6	4136	80.3
22	1013	19.7	5149	100.0

Frequency Missing = 2

Remarque : il est possible d'écrire une ligne de restriction à l'intérieur de la procédure.

```
proc freq data=b; tables sexe*_commod ;
where viecou="1";
run;
```

```
proc freq data=b; tables sexe*_commod ;
where viecou="2";
run;
```

## Proc format

```
proc format;
value _Age
    00-24='18à25'
    25-29='25à29'
    30-39='30à39'
    40-49='40à49'
    50-59='50à60'
    60-98='60&+'
    99='AgeND';

value $POSIT3_
1='SentUtil'
0,2-9='NSentUtil';

data c (keep=age _posit3); set toto.pif;
_posit3=posit3;
if quest=3 then _POSIT3= CPOSIT3;
if quest=5 then _POSIT3= FPOSIT3 ;
run;

proc freq data=c; tables _posit3*age;
format age _age. _posit3 $posit3_.;
run;
```

## Résultat

TABLE OF \_POSIT3 BY AGE

_POSIT3	AGE						Total
	18à25	25à29	30à39	40à49	50à60	60&+	
Frequency	164	371	931	1022	638	107	3233
Percent	4.28	9.68	24.30	26.67	16.65	2.79	84.37
Row Pct	5.07	11.48	28.80	31.61	19.73	3.31	
Col Pct	81.19	81.00	83.13	84.74	88.00	88.43	
NSentUtil	38	87	189	184	87	14	599
	0.99	2.27	4.93	4.80	2.27	0.37	15.63
	6.34	14.52	31.55	30.72	14.52	2.34	
	18.81	19.00	16.88	15.26	12.00	11.57	

Total	202	458	1120	1206	725	121	3832
	5.27	11.95	29.23	31.47	18.92	3.16	100.00

Frequency Missing = 1319

## Proc univariate

```

data d (keep=salaire age homme); set toto.pif;
if quest=1 then _salaire=asalaire;
if quest=2 then _salaire=tsalaire;
if asalfreq="A" then salaire=_salaire/12;
if asalfreq="M" then salaire=_salaire;
if asalfreq="" then salaire=_salaire;
if asalfreq="0" then salaire=.;
if _salaire=999999 then salaire=.;
homme=(sexe=1);

```

```
proc univariate data=d; var salaire; run;
```

## Résultats

Univariate Procedure

Variable=SALAIRE

### Moments

N	2799	Sum Wgts	2799
Mean	9241.963	Sum	25868254
Std Dev	8216.05	Variance	67503478
Skewness	10.72349	Kurtosis	241.9766
USS	4.279E11	CSS	1.889E11
CV	88.89941	Std Mean	155.2965
T:Mean=0	59.51173	Pr> T	0.0001
Num ^= 0	2798	Num > 0	2798
M(Sign)	1399	Pr>= M	0.0001
Sgn Rank	1957901	Pr>= S	0.0001

### Quantiles (Def=5)

100% Max	240000	99%	37000
75% Q3	11000	95%	20000
50% Med	7700	90%	15000
25% Q1	5600	10%	3425
0% Min	0	5%	2500
		1%	1000
Range	240000		
Q3-Q1	5400		
Mode	10000		

### Extremes

Lowest	Obs	Highest	Obs
0 (	3206)	73000 (	2867)
9 (	1754)	83333.17 (	2691)
9 (	1502)	100000 (	347)
130 (	4745)	104000 (	885)
166.6667 (	1062)	240000 (	1813)

Missing Value	.
Count	2352
% Count/Nobs	45.66

## Proc means

```
proc means data=d; var salaire; run;
```

## Résultats

```
The SAS System                               17:37 Monday, November 24, 1997  31
Analysis Variable : SALAIRE
```

N	Mean	Std Dev	Minimum	Maximum
2799	9241.96	8216.05	0	240000.00

## Proc reg :

**procédure pour faire des régressions.** Niveau statistique plus élevé.

```
proc reg data=d; model salaire=age homme;
run;
```

## Résultat

```
The SAS System                               17:37 Monday, November 24, 1997  32
Model: MODEL1
Dependent Variable: SALAIRE

Analysis of Variance
```

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	2	11439200315	5719600157.4	90.129	0.0001
Error	2796	177435531901	63460490.665		
C Total	2798	188874732215			

Root MSE	7966.20930	R-square	0.0606
Dep Mean	9241.96287	Adj R-sq	0.0599
C.V.	86.19608		

```
Parameter Estimates
```

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob >  T
INTERCEP	1	2762.489635	628.84294635	4.393	0.0001
AGE	1	121.863168	14.79595221	8.236	0.0001
HOMME	1	3140.182847	301.22527292	10.425	0.0001

## Proc logistic

**procédure pour faire des régressions logistiques.** Niveau statistique plus élevé.

```
data e (keep=galer revtot1-revtot4 revtot9 sexe1 sexe2
lagel1-lage6 diplom0-diplom6 diplom9); set toto.pif;
if quest=5 then galer=Fgaler;
if galer=9 then galer="";
revtot1=(revtot<6);
revtot2=(6<=revtot<=8);
revtot3=(9<=revtot<=11);
revtot4=(12<=revtot<=13);
```

```

revtot9=(revtot>=14);

label revtot1='pauvres';
label revtot2='moyens';
label revtot3='aise';
label revtot4='riches';
label revtot9='revnrp';

sexe1=(SEXE=1);
sexe2=(SEXE=2);

label SEXE1='Homme';
label SEXE2='Femme';
LAGE1=(AGE<=24);
LAGE2=(25<=AGE<=29);
LAGE3=(30<=AGE<=39);
LAGE4=(40<=AGE<=49);
LAGE5=(50<=AGE<=59);
LAGE6=(60<=AGE);

label LAGE1='0à24';
label LAGE2='25à29';
label LAGE3='30à39';
label LAGE4='40à49';
label LAGE5='50à59';
label LAGE6='60&plus';

DIPLOM0=(DIPLOM=0 or diplom=01);
DIPLOM1=(DIPLOM=11 or diplom=12);
DIPLOM2=(DIPLOM=13);
DIPLOM3=(DIPLOM=21);
DIPLOM4=(DIPLOM=22 or DIPLOM=23);
DIPLOM5=(DIPLOM=30);
DIPLOM6=(DIPLOM=41 or Diplom=42);
DIPLOM9=(DIPLOM=9);

label DIPLOM0='Aucun/CE';
label DIPLOM1='CAP/BEP';
label DIPLOM2='BEPC';
label DIPLOM3='BacTech';
label DIPLOM4='BacGen';
label DIPLOM5='Bac+2';
label DIPLOM6='Supérieur';
label DIPLOM9='DiploNSP';
run;
proc logistic data=e;
model galer= revtot2-revtot4 revtot9 sexe1
lage2-lage6 diplom1-diplom6 diplom9;
run;

```

## Résultat

Data Set: WORK.E  
 Response Variable: GALER  
 Response Levels: 2  
 Number of Observations: 3393  
 Link Function: Logit

### Response Profile

Ordered Value	GALER	Count
1	1	609
2	2	2784

WARNING: 1758 observation(s) were deleted due to missing values for the response or explanatory variables.

### Testing Global Null Hypothesis: BETA=0

Criterion	Intercept Only	Intercept and Covariates	Chi-Square for Covariates
AIC	3195.593	3166.190	.
SC	3201.723	3270.390	.
-2 LOG L Score	3193.593	3132.190	61.404 with 16 DF (p=0.0001) 60.683 with 16 DF (p=0.0001)

NOTE: The following parameters have been set to 0, since the variables are a linear combination of other variables as shown.

DIPLOM9 = 0

### Analysis of Maximum Likelihood Estimates

Variable	DF	Parameter Estimate	Standard Error	Wald Chi-Square	Pr > Chi-Square	Standardized Estimate	Odds Ratio
INTERCPT	1	-0.8679	0.2209	15.4311	0.0001	.	.
REVTOT2	1	-0.2336	0.1373	2.8921	0.0890	-0.061688	0.792
REVTOT3	1	-0.3573	0.1430	6.2443	0.0125	-0.095944	0.700
REVTOT4	1	-0.1840	0.2144	0.7366	0.3907	-0.028151	0.832
REVTOT9	1	-0.3644	0.2579	1.9972	0.1576	-0.040510	0.695
SEXE1	1	0.3754	0.0916	16.8077	0.0001	0.102868	1.456
LAGE2	1	-0.2936	0.2322	1.5988	0.2061	-0.051692	0.746
LAGE3	1	-0.1947	0.2070	0.8842	0.3470	-0.048843	0.823
LAGE4	1	-0.3062	0.2066	2.1959	0.1384	-0.078993	0.736
LAGE5	1	-0.5777	0.2219	6.7787	0.0092	-0.125519	0.561
LAGE6	1	-0.3150	0.3217	0.9587	0.3275	-0.029656	0.730
DIPLOM1	1	-0.2298	0.1203	3.6504	0.0561	-0.057468	0.795
DIPLOM2	1	-0.5197	0.1908	7.4152	0.0065	-0.078845	0.595
DIPLOM3	1	-0.5552	0.2372	5.4801	0.0192	-0.066606	0.574
DIPLOM4	1	-0.3464	0.1888	3.3650	0.0666	-0.051954	0.707
DIPLOM5	1	-0.2951	0.1621	3.3151	0.0686	-0.054138	0.744
DIPLOM6	1	-0.7891	0.1887	17.4794	0.0001	-0.141076	0.454
DIPLOM9	0	0	.	.	.	.	.

### Association of Predicted Probabilities and Observed Responses

Concordant = 59.0%	Somers' D = 0.197
Discordant = 39.3%	Gamma = 0.200
Tied = 1.7%	Tau-a = 0.058
(1695456 pairs)	c = 0.598

## Proc corresp

Procédure pour faire des analyses factorielles. Niveau statistique élevé.

```
proc format;
```

```

value $AVENT
1='Aventure'
2,9='NAventur';

value $EQUIP
1='Equip'
2,9='NEquip';

value $GALER
1='Galer'
2,9='NGaler';

value $JUNGL
1='Jungle'
2,9='NJungle';

value $OBSTA
1='Obsta'
2,9='NObsta';

value $PASSI
1='Passi'
2,9='NPassi';

value $ROUTI
1='Routi'
2,9='NRouti';

value $VGARA
1='VGara'
2,9='NVgara';

value $SEXE
    1='Homme'
    2='Femme';

value $DIPLO
    0,9='Aucun/CE'
    1='CAP-BEP (C)'
    2='BAC'
    3='BAC+2'
    4='Supérieur';

value $CS6_
    10-19,71='Agri'
    20-22,72='ArtCom'
    23,30-39,74='Cadres'
    40-47,75='P.Int'
    50-59,77='Emplo'
    48,60-69,78='Ouv'

```

```

80-89='Inact'
99='CSND';

run;

data f; set toto.pif;
if quest=1;
options ps=75;

proc corresp data=f MCA OUTC=COORD DIMENS=2;
tables GALER ROUTI VGARA PASSI AVENT OBSTA JUNGL EQUIP CS
SEXE DIPLO ;
supplementary CS SEXE DIPLO ;
FORMAT galer $GALER. routi $routi. vgara $vgara. passi
$passi. avent $avent.
Obsta $OBSTA. jungl $jungl. CS $CS6_. sexe $sexe. diplo
$diplo. equip $equip.
;
run;

Proc plot data=coord;
plot dim2*dim1$_NAME_/HREF=0 VREF=0;
run;

```

The SAS System

17:37 Monday, November 24, 1997 37

The Correspondence Analysis Procedure

Inertia and Chi-Square Decomposition

Singular Values	Principal Inertias	Chi-Squares	Percents	4	8	12	16	20
0.47199	0.22277	5690.25	22.28%	*****	*****	*****	*****	*****
0.44578	0.19872	5075.88	19.87%	*****	*****	*****	*****	*****
0.34649	0.12005	3066.57	12.01%	*****	*****	*****	*****	*****
0.32589	0.10621	2712.83	10.62%	*****	*****	*****	*****	*****
0.31166	0.09713	2481.06	9.71%	*****	*****	*****	*****	*****
0.30036	0.09022	2304.38	9.02%	*****	*****	*****	*****	*****
0.28885	0.08343	2131.13	8.34%	*****	*****	*****	*****	*****
0.28543	0.08147	2080.98	8.15%	*****	*****	*****	*****	*****
	1.00000	25543.1	(Degrees of Freedom = 225)					

Column Coordinates

	Dim1	Dim2
Galer	1.09403	1.02963
NGaler	-0.24846	-0.23384
NRouti	-0.50409	0.28589
Routi	0.78513	-0.44529
NVgara	-0.17982	-0.07810
VGara	1.82678	0.79345
NPassi	0.67974	-0.32022
Passi	-0.54774	0.25803
Aventure	-0.65947	0.72120
NAventur	0.35516	-0.38840
NObsta	-0.05299	-0.57346
Obsta	0.07105	0.76887
Jungle	0.72158	1.41765
NJungle	-0.11827	-0.23235
Equip	-0.17236	-0.09356

NEquip 0.52474 0.28484

Supplementary Column Coordinates

	Dim1	Dim2
Agri	0.050651	0.416688
ArtCom	-.076168	0.531705
CSND	-.024332	0.156689
Cadres	-.325315	0.269711
Emplo	0.129721	-.189163
Ouv	0.295975	-.242491
P.Int	-.271044	0.077760
Femme	0.008869	-.077432
Homme	-.008488	0.074106
Aucun/CE	0.307552	-.192298
BAC	-.074753	0.067829
BAC+2	-.247148	0.160510
CAP-BEP(C)	0.087148	-.055479
Supérieur	-.410676	0.232004

The SAS System

17:37 Monday, November 24, 1997 38

Summary Statistics for the Column Points

	Quality	Mass	Inertia
Galer	0.512592	0.023135	0.101865
NGaler	0.512592	0.101865	0.023135
NRouti	0.523076	0.076125	0.048875
Routi	0.523076	0.048875	0.076125
NVgara	0.390454	0.113798	0.011202
VGara	0.390454	0.011202	0.113798
NPassi	0.454949	0.055779	0.069221
Passi	0.454949	0.069221	0.055779
Aventure	0.514333	0.043755	0.081245
NAventur	0.514333	0.081245	0.043755
NObsta	0.444680	0.071598	0.053402
Obsta	0.444680	0.053402	0.071598
Jungle	0.414735	0.017602	0.107398
NJungle	0.414735	0.107398	0.017602
Equip	0.117095	0.094093	0.030907
NEquip	0.117095	0.030907	0.094093

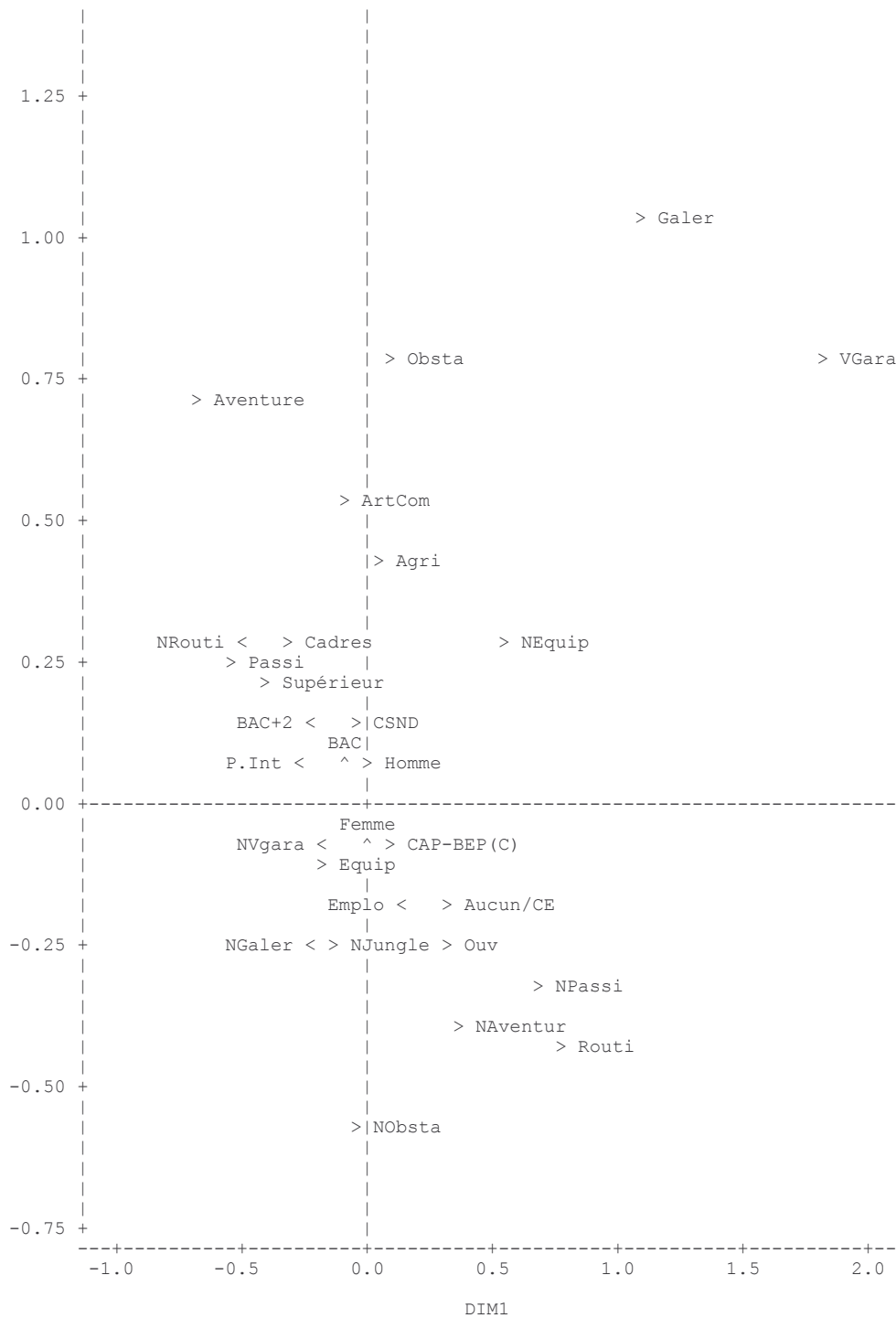
Quality of Representation for the Supplementary Column Points

Agri	0.006689
ArtCom	0.023017
CSND	0.000345
Cadres	0.029623
Emplo	0.020292
Ouv	0.046962
P.Int	0.021457
Femme	0.005813
Homme	0.005813
Aucun/CE	0.035582
BAC	0.001550
BAC+2	0.014320
CAP-BEP(C)	0.006603
Supérieur	0.033630

Partial Contributions to Inertia for the Column Points

	Dim1	Dim2
Galer	0.124297	0.123419
NGaler	0.028229	0.028030
NRouti	0.086831	0.031311
Routi	0.135242	0.048768
NVgara	0.016517	0.003493
VGara	0.167800	0.035488
NPassi	0.115691	0.028782
Passi	0.093225	0.023193
Aventure	0.085419	0.114526
NAventur	0.046002	0.061678
NObsta	0.000902	0.118487





NOTE: 1 obs had missing values.

### Exemple de programme SAS :

```
libname compil 'c:\sas\macro';
```

```

options sasstore=compil mstored nodate nonumber ps=59;

data a; set toto.argent;

cs6csp6=100*cs+csp;
cs6reli=cs+100*relig;
if cs>80 then cs=.;
if _aspec3=9 then _aspec3=.;
if _ipourq3=9 then _ipourq3=.;
if _rester=9 then _rester=.;

proc format;
value cs6csp6z
1010-1019='1F1'
1020-1099='1FAut'
2110-2131,2210-2231='2FInd'
2133-2199,2233-2299='2FAut'
2310-2323,3110-3123,3710-3723,3810-3823='36Find'
3310-3323,3410-3423,3510-3523='32FInd'
3330-3339,3430-3439,3530-3539='32F3'
2330-2339,3130-3139,3730-3739,3830-3839='36F3'
3340-3399,3440-3499,3540-3599='32FSal<'
2340-2399,3140-3199,3740-3799,3840-3899='36FSal<'
4231-4239,4331-4339,4431-4439,4531-4539='41F3'
4210-4223,4240-4299,4310-4323,4340-4399,4410-4423,4440-4499,4510-4523,4540-
4599,
4640-4699,4710-4759,4810-4859='4FAut'
4610-4623 ='46FInd'
4630-4639 ='46F3'
4760-4799,4860-4899='47F6'
5230-5259,5330-5359,5430-5459,5530-5559,5630-5659='5FSa>'
5210-5223,5310-5323,5410-5423,5510-5523,5610-5623='5FInd'
5260-5299,5360-5399,5460-5499,5560-5599,5660-5699='5F6'
6230-6259,6330-6359,6430-6459,6530-6559='61FSal>'
6210-6223,6310-6323,6410-6423,6510-6523='61FInd'
6260-6299,6360-6399,6460-6499,6560-6599='61F6'
6730-6759,6830-6859,6930-6959='66FSal>'
6710-6723,6810-6823,6910-6923='66FInd'
6760-6799,6860-6899,6960-6999='66F6';

value cs6reliz
110,910='1NR'
210='1NP'
310='1R'
121-122,921-922='2NR'
221-222='2NP'
321-322='2R'
123-131,137-139,923-931,937-939='36NR'
223-231,237-239='36NP'
323-331,337-339='36R'
132-135,932-935='32NR'
232-235='32NP'
332-335='32R'
141-145,941-945='41NR'
241-245='41NP'
341-345='41R'
146,946='46NR'
246='46NP'
346='46R'
147-148,947-948='47NR'

```

```

247-248='47NP'
347-348='47R'
151-154,951-954='5QNR'
251-254='5QNP'
351-354='5QR'
155-159,955-959='55NR'
255-259='55NP'
355-359='55R'
161-165,961-965='61NR'
261-265='61NP'
361-365='61R'
167-169,967-969='66NR'
267-269='66NP'
367-369='66R';

```

```

value diprevz
1='Di0Re1'
2='Di0Re2'
3-5='Di0Re35'
11-12='Di1Re12'
13='Di1Re2'
14-15='Di1Re45'
21-23='Di2Re13'
24-25='Di2Re45'
31-33='Di3Re13'
34-35='Di3Re45'
41-43='Di4Re13'
44='Di4Re4'
45='Di4Re5'
99='Di9/Re9';

```

```

value HREVENUZ
0,9='NHRevenu'
1='HRevenu';

```

```

value RREVENUZ
0,9='NRRevenu'
1='RRevenu';

```

```

proc corresp noprint Data=A Outc=Sor dimens=4;
tables ident, httarg rttarg pttarg _paye _ipourq3 _posit2 _negat1 _sala
_rester _aspec3
cs age salair3 sexe
diprev cs6csp6 cs6reli
;
supplementary
cs age salair3 sexe
diprev cs6csp6 cs6reli
;
format httarg httargz. rttarg rttargz. pttarg pttargz. _paye _payez.
_ipourq3 ipourq3z. _posit2 _posit2z. _negat1 _negat1z. _sala _salaz.
_rester _resterz. _aspec3 _aspec3z.
cs cs82Z. age agez. salair3 salair7z. sexe sexez.
diprev diprevZ. cs6csp6 cs6csp6z. cs6reli cs6reliz.
;

```