

REPORTING avec SAS

Petites astuces pratiques :

1. Pour éviter d'avoir les numéros de pages et la date du jour dans votre document WORD (ou PDF ou autre), exécutez cette ligne (n'importe où et une seule fois suffit pour toute la session SAS) :

```
options nodate nonumber ;
```

2. Pour éviter à l'utilisateur de devoir entrer dans votre macro pour changer les noms des répertoires, penser à les déclarer en macro-variables avant le début de la macro, par exemple :

```
%let rep ="z:\votre répertoire " ;  
%macro acp(table, variables, identifiant) ;
```

...

L'utilisateur n'aura donc que le chemin à actualiser sans entrer dans la macro.

3. Même conseil pour les librairies, déclarez-les en dehors de votre macro :

```
libname malib "z:\votre répertoire " ;  
%macro acp(table, variables, identifiant) ;
```

-

Proc REPORT

La **PROC REPORT** (de **REPORTing**), permet de mettre en forme les sorties SAS pour créer des rapports personnalisés (Couleur, Police, forme, ...), de créer de nouvelles variables synthétiques (statistiques ou autres), ... Elle s'approche, un peu, de la proc **TABULATE** mais propose une grande souplesse dans l'ajustement des styles (en lien avec l'ODS). Nous présentons ici les premières bases de cette procédure qui nous seront utiles dans cette SAE.

1. Proc report, une proc print personnalisée :

Dans sa syntaxe de base, sans aucune option, la **proc report** a exactement le même effet qu'une **proc print**. Ainsi, les deux programmes suivants sont équivalents :

Report	Print
<i>PROC REPORT DATA</i> = table SAS <i>NOWD</i> ; <i>COLUMNS</i> liste des variables à afficher ; <i>RUN</i> ;	<i>PROC PRONT DATA</i> = table SAS ; <i>VAR</i> liste des variables à afficher ; <i>RUN</i> ;

Dans la procédure **REPORT**, l'instruction **COLUMNS** est utilisée pour sélectionner les colonnes avec lesquelles on veut travailler (elle joue le même rôle de **VAR** dans la proc print et d'autres procédures SAS) et **NOWD** (No WinDow) est là pour empêcher SAS d'ouvrir une fenêtre interactive (l'omission de cette option n'est pas gênant dans les dernières versions de SAS).

2. Mise en forme des colonnes, **DEFINE** :

L'instruction **DEFINE** permet de personnaliser l'affichage des colonnes, sa syntaxe est la suivante :

DEFINE nom_variable / **OPTIONS** ;

- Quelques options :

a. Libellés à afficher :

DEFINE nom_variable / **DISPLAY** "texte à afficher " ;

Exemple :

```
proc report data=notes nowd;

/* On sélectionne les variables avec lesquelles on veut
travailler */
    columns nom sexe math ;

/* On définit les libellés à afficher */
    define nom / display "Nom de l'étudiant";
    define sexe / display "Sexe de l'étudiant";
    define math / display "Note en math";

run;
```

Nom de l'étudiant	Sexe de l'étudiant	Note en math
E1	M	6
E2	F	16
...

b. Format d'affichage :

DEFINE variable / **DISPLAY** "texte à afficher" **FORMAT=** format;

Le format pouvant être un **format SAS standard** ou un **format personnalisé**.

Exemples :

▪ Format SAS :

```
proc report data=notes ;
  columns nom sexe math ;
  define nom / display "Nom de l'étudiant";
  define sexe / display "Sexe de l'étudiant";
  define math / display "Note en math" format=6.2;
run;
```

Nom de l'étudiant	Sexe de l'étudiant	Note en math
E1	M	6.00
E2	F	16.00
...

▪ Format personnalisé

- Création du format

```
proc format ;
  value fmt
    0-8 ="Faible"
    8-12="Moyen"
    12-high = "Bon"
  ;
run;
```

- Application du format

```
Define math / display "Note en math" format=fmt. ;

/* Ne pas oublier le point après le nom du format */
```

Nom de l'étudiant	Sexe de l'étudiant	Note en math
E1	M	Faible
E2	F	Bon
E3	M	Moyen
...

c. Mise en forme de l'entête :

DEFINE variable / **DISPLAY** "texte à afficher " **FORMAT=** format

STYLE (HEADER) = [options2] ;

- Quelques options2 :

○ Couleurs (FOND et TEXTE) :

[background=Couleur du fond foreground=Couleur du texte]

Exemple :

```
define nom / display "Nom de l'étudiant"

style (header)=[background=cyan      foreground=red] ;
```

Nom de l'étudiant	sexe de l'étudiant	Note en math
E1	M	Faible
E2	F	Bon
E3	M	Moyen
...

- Police et taille et attributs de la cellule :

Options	Valeurs
FONT_FACE =	" nom d'une Police de caractère "
FONT_SIZE =	Taille en points (Taille WORD=6.5+1.5*taille_SAS)
FONT_WEIGHT=	BOLD (gras) MEDIUM (Normal)
just= (just pour justify)	C (centré) L (aligné à gauche) R (aligné à droite)
...	...

NB.

Quand on travaille avec un ODS RTF, on peut utiliser les polices de WORD.

Exemple :

```
define nom / display "Nom de l'étudiant"
Style(header) = [FONT_FACE="French Script MT"
FONT_SIZE= 10 FONT_WEIGHT=bold just= c];
```

Nom de l'étudiant	sexe	Note en math
E1	M	Faible
E2	F	Bon

- d. Mise en forme des colonnes (autre que l'entête) :**

STYLE (COLUMN) = [les mêmes options que plus haut]

Exemple, pour la colonne nom :

```
define nom / display "Nom de l'étudiant"
    style(header) = [background=cyan foreground=red
    FONT_FACE="French Script MT" FONT_SIZE= 10
    FONT_WEIGHT=bold          just= c]

style(column) = [background=yellow foreground=green
just=c FONT_SIZE= 5] ;
```

Nom de l'étudiant	sexe	Note en math
E1	M	Faible
E2	F	Bon

e. Utiliser un format conditionnel pour la couleur du fond :

Au lieu de faire un format uniforme pour toute une colonne, on peut faire un format en fonction des valeurs de la variable.

Création des formats :

```
proc format ;
value note
    0-8 =red
    8-12=orange
    12-high = green
;
Value $sexe
    "M"=blue
    "F"=magenta
;
run;
```

Utilisation :

```
define math / display "Note en math" format=fmt.
```

```
style(column)=[ background =note.];
```

```
Define sexe / display "Sexe de l'étudiant"
```

```
style(column)=[ background =$sexe.];
```

Nom de l'étudiant	Sexe de l'étudiant	Note en math
E1	M	faible
E2	F	bon
E3	M	moyen

Remarque :

Dans ce qui précède, nous avons mis en forme, une à une toutes les colonnes. Les caractéristiques communes telles la police, la forme de l'entête, ... peuvent être regroupées dans la définition du rapport :

```
proc report data=Notes
```

```
/* On définit les caractéristiques communes aux colonnes, un seul point-virgule plus bas*/
```

```
style(header)=[color=green          background=cornflowerblue
font_face="French Script MT"        fontsize=30pt
fontweight=bold just=c]
style(column)=[color=blue  fontsize=20pt  bordercolor=blue
just=c ]
```

```
;
```

```
columns nom sexe math ;
```

```
/* On définit les caractéristiques NON communes */
```

```
define nom / display "Nom de l'étudiant" ;
```

```
define math / display "Note en math" format=fmt.
```

```
style(column)=[foreground=white background =note.]
```

```
;
```

```
Define sexe / display "Sexe de l'étudiant"
```

```
style(column)=[foreground=white background =$sexe.];
```

```
run;
```

Nom de l'étudiant	Sexe de l'étudiant	Note en math
E1	M	faible
E2	F	bon
E3	M	moyen

Sous Word on obtient :

Nom de l'étudiant	Sexe de l'étudiant	Note en math
E1	M	faible
E2	F	bon
E3	M	moyen

Remarque :

Certains libellés peuvent être trop longs ('Nom de l'étudiant', par exemple) et donnent à l'affichage des colonnes trop larges. On peut imposer à SAS les emplacements des retours à la ligne avec la commande ***SPLIT*** :

Exemple :

Dans l'exemple ci-dessous, il reviendra à la ligne chaque fois qu'il rencontrera un # (on peut choisir n'importe quel caractère) :


```

proc report data=Notes SPLIT=' # '

style(header)=[color=green background=cornflowerblue
font_face="French Script MT"
font_size=25pt fontweight=bold just=c]
style(column)=[color=blue font_size=20pt bordercolor=blue
just=c ];

columns num sexe math ;
define num / display "Nom # de # l'étudiant" ;

define math / display "Note # en # math" format=fmt.
style(column)=[foreground=white background =note.]
;

Define sexe / display "Sexe " ;

run;

```

Nom de l'étudiant	Sexe	Note en math
E1	M	Bon
E2	F	Bon
E3	M	Moyen

3. Calcul de Nouvelles colonnes dans proc report : **COMPUTE**

Outre les fonctions standards (moyenne, somme, ...) que nous n'aborderons pas ici, on peut calculer de nouvelles variables personnalisées. Dans l'exemple ci-dessous, nous allons calculer la note en math sur 30 et ajouter un bonus de 3 points si cette note est inférieure à 15.

La nouvelle variable s'appellera math30 et il faut **la déclarer dans COLUMNS** et pour sa mise en forme préciser qu'elle est calculée (**COMPUTED**).

```

proc report data=Notes  split='#'
style(header)=[color=green background=cornflowerblue
font_face="French Script MT"
fontsize=25pt fontweight=bold just=c]
style(column)=[color=blue fontsize=20pt bordercolor=blue
just=c ];

columns num sexe math math30;
define num / display "Nom # de # l'étudiant" ;
define math / display "Note # en # math" ;
Define sexe / display "Sexe " ;

Define math30 / computed "note # sur # 30 " ;

compute math30;
math30 = math*3/2 ;
if math30 <15 then math30 =math30+3 ;
endcomp;

run;

```

Nom de l'étudiant	Sexe	Note en math	note sur 30
E1	M	16	24
E2	F	16	24
...
E19	F	8	15

Pour vous entrainer avant de commencer

Les exemples à suivre ne sont pas à rendre mais juste pour vous entrainer et vous familiariser avec la procédure **REPORT**. Il est vivement conseillé de prendre le temps de les refaire, cela vous fera gagner beaucoup de temps lors de la réalisation de la SAE.

Exemple 1 : un premier exemple simple

Dans cet exemple, nous allons réaliser un test du χ^2 , récupérer la valeur du V de Cramer, l'interpréter et rajouter du texte dans la sortie RTF (nous travaillons avec le fichier Titanic). Plus précisément, nous voulons avoir ce document WORD :

Statistique	Valeur	Commentaire
V de Cramer	0.3987	Liaison très forte

a) Récupérer le V de Cramer dans une table :

```
ods exclude all;
ods output chisq =chi2;
proc freq data=titanic;
table sexe*classe /chisq ;
run;
ods select all;
```

Une image de la table chi2 créée par cette procédure est donnée ci-dessous :

Table	Statistique	DDL	Valeur	Prob
Table sexe * classe	Khi-2	3	349.9145	<.0001
Table sexe * classe	Test du rapport de vraisemblance	3	412.6012	<.0001
Table sexe * classe	Khi-2 de Mantel-Haenszel	1	278.9396	<.0001
Table sexe * classe	Coefficient Phi	—	0.3987	—
Table sexe * classe	Coefficient de contingence	—	0.3704	—
Table sexe * classe	V de Cramer	—	0.3987	—

Nous récupérons la ligne concernant le 'V de Cramer' ici sur la version française. Dans la version anglaise, il s'appelle 'Cramer's V', voir plus bas :

Table	Statistic	DF	Value	Prob
Table sexe * classe	Chi-Square	3	349.9145	<.0001
Table sexe * classe	Likelihood Ratio Chi-Square	3	412.6012	<.0001
Table sexe * classe	Mantel-Haenszel Chi-Square	1	278.9396	<.0001
Table sexe * classe	Phi Coefficient	–	0.3987	–
Table sexe * classe	Contingency Coefficient	–	0.3704	–
Table sexe * classe	Cramer's V	–	0.3987	–

Pour pouvoir rajouter le commentaire, nous dupliquons la colonne 'Value'.

```
data v ;
set chi2;
where statistic="V de Cramer";
comment=Value; /* On duplique la colonne Value */
keep statistic value comment;
run;
```

Nous créons un format d'affichage et un format de couleur (plus la liaison sera forte, plus la couleur sera sombre) :

```
proc format ;
value comm
0.00      ="Aucune relation"
.00 - .15 ="Liaison très très faible"
.15 - .20 ="Liaison faible"
.20 - .25 ="Liaison modérée"
.25 - .30 ="Liaison assez forte"
.30 - .35 ="Liaison forte"
.35 - .40 ="Liaison très forte"
.40 - .45 ="Liaison tROP forte"
.45 - .99 ="Redondance"
1.0      ="Relation parfaite"
;

value coul
0.00-0.20 = VLIGB
0.20-0.30 =BIGB
other = BIPB
;
/* Pour simplifier, nous n'avons pris que trois niveaux de
couleur */
run;
```

Enfin la proc report :

```
ods rtf file="c:\aq\cramer.rtf" STARTPAGE=NO;
proc report data=v ;
column statistic value comment;
define comment / display "Commentaire" format =comm.
style(column)=[background=coul. foreground=white] ;
run;
ods rtf close;
```

Nous obtenons le document Word :

Statistique	Valeur	Commentaire
V de Cramer	0.3987	Liaison très forte

Il n'y a plus qu'à paramétrer pour en faire une macro !

Exemple 2 :

Le but de cet exemple est d'écrire une macro qui réalise la RLS d'une variable Y sur une autre variable X et qui teste la validité du modèle.

- a) On écrit un programme pour math et stat et une fois terminé il ne restera plus qu'à le paramétrer pour qu'il fonctionne sur toutes les tables et toutes les variables.

```
/* On supprime tous les sorties, on n'en a pas besoin */
ods exclude all;
```

```
/* On récupère le tableau d'ANOVA dans tab_anova */
ods output anova=tab_anova;
```

```
/* On réalise la proc reg */
proc reg data=notes ;
model math=stat ;
run;
quit;
```

```
/* On rétablit l'affichage */
ods select all;
```

La table tab_anova se présente comme ci-dessous :

Model	Dependent	Source	DDL	Somme des carrés	Moyenne quadratique	Valeur F	Pr > F
MODEL1	math	Modèle	1	254.68494	254.68494	23.47	<.0001
MODEL1	math	Erreur	48	520.83506	10.85073	—	—
MODEL1	math	Total sommes corrigées	49	775.52000	—	—	—

a) On veut surligner la colonne p-valeur en :

- Cyan si la p_valeur est inférieure à 1%
- Jaune si la p_valeur est comprise entre 1% et 5%

Création du format :

```
proc format ;
value pval
    0-0.01 =cyan
    0.01-0.05 =yellow
;
run;
```

Proc report :

```
ods rtf file="c:\aq\rls.rtf" STARTPAGE=NO;
proc report data =tab_anova
    style(header)=[color=black background=cornflowerblue
font_face="Times New Roman"
    fontsize=15pt fontweight=bold just=c]
    style(column)=[color=black fontsize=15pt bordercolor=blue
just=c ];
columns source -- probf ;
define source / display " Source"
style(column)=[font_face="Times New Roman" fontweight=bold];
define DF / display "DDL";
define SS / display "SCE" format =8.2;
define MS / display "CM" format =8.2;
define FValue / display "F calculé" format =8.2;
define probf / display "P-valeur" format =8.3
style(column)=[background=pval.];
run;
ods rtf close ;
```

Résultat sur Word :

Source	DDL	SCE	CM	F calculé	P-valeur
Modèle	1	254.68	254.68	23.47	0.000
Erreur	48	520.84	10.85	—	—
Total sommes corrigées	49	775.52	—	—	—

b) On veut maintenant, afficher un commentaire : ‘Modèle validé à 1%’ ou ‘Modèle validé à 5%’ ou ‘Modèle NON validé à 5%’

Contrairement à l’exemple 1, nous allons écrire le commentaire à la place de la p-valeur.

On commence par créer un format pour le commentaire :

```
proc format ;
value comm
    0-0.01 ="Modèle validé à 1%"
    0.01-0.05 ="Modèle validé à 5%"
    0.05-high ="Modèle Non validé à 5%"
;
run;
```

La ligne concernant la p-valeur devient :

```
define probf / display "Décision" format =comm.
style (column)=[background=pval. ] ;
```

Et on obtient sous Word :

Source	DDL	SCE	CM	F calculé	Décision
Modèle	1	254.68	254.68	23.47	Modèle validé à 1%
Erreur	48	520.84	10.85	—	—
Total sommes corrigées	49	775.52	—	—	—

c) Maintenant nous allons nous occuper du R^2 :

Nous allons juste le récupérer dans une macro-variable (MV) et l’ajouter, dans un texte à part, dans la sortie Word.

Pour ajouter un texte dans une sortie ODS (RTF, PDF) :

ODS TEXT = " texte à afficher " ;

- Récupération du R^2 :

Avec l'ODS, on peut récupérer plusieurs tables en même temps, la proc reg devient (Tab2 contiendra le R^2) :

```
ods exclude all;
ods output anova=tab_anova      FitStatistics = tab2;
proc reg data=notes ;
model math=stat ;
run;
quit;
ods select all;
```

La table Tab2 se présente sous la forme :

Dependent	Label1	cValue1	nValue1	Label2	cValue2	nValue2
math	Root MSE	3.29404	3.294045	R carré	0.3284	0.328405
math	Moyenne dépendante	12.36000	12.360000	R car. ajust.	0.3144	0.314414
math	Coeff Var	26.65085	26.650847			0

cValue = Charcter Value (format alphanumérique)

nValue = Numeric Value (format numérique)

On récupère le R^2 (nvalue2) dans une MV, plus exactement $100 \times R^2$ arrondi au centième près :

```
data _null_ ;
set tab2;
if Label2 = 'R carré' then call
    symputx('R2', round(100*nValue2, 0.01));
run;
```

NB.

Adaptez si vous avez la version anglaise.

On affiche le R^2 dans la sortie RTF :

```
ods text=" "; /* pour créer un saut de ligne */
ods text="La note en stat explique &R2 % des variations de la
note en math";
```


Sortie Word :

Source	DDL	SCE	CM	F calculé	Décision
Modèle	1	254.68	254.68	23.47	Modèle validé à 1%
Erreur	48	520.84	10.85	—	—
Total sommes corrigées	49	775.52	—	—	—

La note en stat explique 32.84 % des variations de la note en math

Mise en forme du texte ajouté :

On peut aussi mettre en forme le texte à afficher. Il existe plusieurs méthodes. La plus simple est d'utiliser (encore) proc report :

```
/* On crée une table SAS ne contenant que le texte à afficher
*/
```

```
data texte ;
texte="La note en stat explique &R2 % des variations de la
note en math" ;
run;
proc report data=texte style(report)=[borderwidth=0] ;
/* borderwidth=0 pour ne pas encadrer */
column texte ;
define texte /display "" style(column)=[font_face="Times New
Roman" fontweight=bold foreground=blue fontsize=6] ;
run;
```

Et on obtient :

La note en stat explique 32.84 % des variations de la note en math

Maintenant, il ne reste plus qu'à paramétrer pour avoir une macro qui fonctionne sur toutes les tables et toutes les variables. Par exemple, si je veux faire la RLS de la vitesse d'une voiture sur sa puissance (la table SAS s'appelle voitures) :

```
%reg(voitures,puissance,vitesse) ;
```

Nous donne :

Source	DDL	SCE	CM	F calculé	Décision
Modèle	1	1786.44	1786.44	39.74	Modèle validé à 1%
Erreur	16	719.17	44.95	—	—
Total sommes corrigées	17	2505.61	—	—	—

La variable puissance explique 71.3 % des variations de vitesse

Encadrement des colonnes : BORDER

Ci-dessus, , nous avons demandé de ne pas l'encadrer (**borderwidth=0**), on peut jouer avec l'encadrement des colonnes :

Borderwidth = épaisseur de l'encadrement

Bordercolor=couleur de l'encadrement

Borderleftwidth = épaisseur du trait de gauche

Borderleftcolor = couleur du trait de gauche

Borderrightwidth = épaisseur du trait de droite

Borderrightcolor = couleur du trait de gauche

Exemple :

```
style(report)=[borderrightwidth=8 Borderrightcolor=red
                borderleftwidth=8 Borderleftcolor=blue];
```

Exemple 3 : Utilisation de Compute pour la mise en forme

Supposons qu'on ait la table suivante :

Num	X	Y
1	26	1200
2	24	1330
3	23	120
4	20	130
5	12	140
6	10	1500

Et qu'on veuille afficher sur SAS :

Num	X	Y
1	36	1200
2	34	1330
3	33	120
4	30	130
5	19	140
6	10	1500

Autrement dit surligner toutes les valeurs de X supérieures à 20 et la cellule de Y correspondant au dernier X surligné et je sais que c'est la 4^{ème} ligne :

```
proc format ;
value coul
20-high =yellow
;
run;

proc report data=table;
columns num x y ;
define num /display "numéro";
define x / display "X" style(column)=[background=coul.];
define y / display "Y";

compute y;
if num = 4 then
call define (_col_, 'style', 'style=[ background=coul.]');
endcomp;
/* on aurait pu mettre if x >20 */
run;
```

Si num = 4 alors on met en forme la colonne Y (COL).

Si à la place de _col_ on met row il surlignera toute la quatrième ligne.

Important :

Il faut que la colonne num soit déclarée avant la colonne Y dans l'instruction COLUMNS.

Exemple 3 :

On peut même faire une boucle dans proc report (quand on est en mode macro) !

Je vous laisse regarder ce petit programme :

```

data a;
input x1 x2 x3 x4 ;
cards;
1 2 3 4
5 3 5 6
2 5 6 8
6 4 7 5
;
run;

proc format ;
value toto
    0-2= blue
    2-5=red
    other=green
;
run;

/* on aurait pu mettre des paramètres dans cette macro mais ce
n'est pas le but de l'exemple */

%macro coloriage;
proc report data=a;
columns x1--x4;
%do i=1 %to 4 ;
define x&i /display "X&i" format=8.3
                style(column)=[ background=toto.];
%end;
run;
%mend ;
%coloriage;

```

Travail à faire

L'objectif de cette SAE est d'écrire une macro ayant **exactement trois paramètres** :

- **Le nom d'une table SAS,**
- **La liste des variables (numériques),**
- **L'identifiant des lignes (alphanumérique).**

et qui revoie un document Word.

Cette macro doit automatiser **la réalisation et l'interprétation** de l'Analyse en Composantes Principales (ACP). Autrement dit, **elle doit faire comme nous faisons à la main** :

- Décider du nombre d'axes à retenir et afficher, dans le document Word, le pourcentage d'inertie restitué par l'ensemble des axes retenus avec une phrase du type :

" Nous avons **tant** de valeurs propres supérieures à 1, nous retenons donc **tant** d'axes. Avec ces axes, nous restituons **tant** % de l'inertie totale. "

Les valeurs propres supérieures à 1, ainsi que le pourcentage cumulé seront, bien entendu, surlignés.

- Faire le coloriage de tous les axes retenus (tableau des coordonnées).
- Faire tous les graphiques des variables et des individus avec **tous axes retenus** (par exemple, si vous retenez 3 axes : il faut faire six graphiques 'axe2*axe1, axe3*axe1 et axe3*axe2' pour les individus et la même chose pour les variables, donc 6 graphiques en tout), vous noterez que l'indice de l'axe en ordonnée est toujours supérieur à celui de l'axe en abscisse).
- Sur ces graphiques, nous adopterons la règle suivante (la même qu'en TD) :
 - o Faire apparaître **en rouge** les éléments (variables ou individus) interprétables sur **l'axe horizontal**.
 - o Faire apparaître **en bleu** les éléments (variables ou individus) interprétables sur **l'axe vertical**.
 - o Faire apparaître **en vert** les éléments interprétables sur **les deux axes** (intersection du bleu et du rouge).
 - o Ne pas afficher les éléments non interprétables.
- Afin de se retrouver dans les graphiques, pensez à mettre des titres.

Vous trouverez en document joint un modèle de sortie attendue.

Cette macro doit fonctionner avec toutes les tables SAS et toutes les variables (numériques) que l'utilisateur sélectionnera (il n'est obligé d'utiliser toutes les variables de la table SAS)

A rendre

- 1. Programme SAS (une seule macro).**
- 2. Un exemple (PDF ou WORD) d'une sortie.**

Pour ceux qui veulent aller plus loin, j'ai déposé un livre (en PDF) : **Reporting avec SAS** (un bon livre que je vous conseille de récupérer).

Avertissement

Il n'est pas interdit de s'entraider, bien au contraire, mais deux macros trop semblables entraînent la même sanction pour les deux groupes.