S'amuser avec les apostrophes et les macros… ou presque!

Par Mathieu Gaouette

Plan

- Le problème
- Contourner le problème
- Faire face au problème
- Références

Le problème

- Analysons comment survient ce problème…
- Roger créé un rapport et il en est fier.
- Il décide de mettre le titre du rapport dans une macro variable pour le changer plus tard.

```
%let titre_rapport = Super rapport de Roger ;

title &titre_rapport. ;

proc freq data=sashelp.cars ;
    table make ;
quit ;
```

Le problème

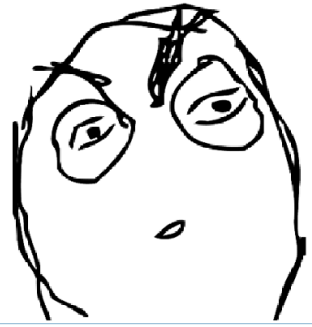
- Roger aime VRAIMENT son rapport.
- Il veut rendre le titre encore plus … incroyable.

```
%let titre_rapport = L'incroyable rapport de Roger ;

title &titre_rapport. ;

proc freq data=sashelp.cars ;
    table make ;
quit ;
```

Log

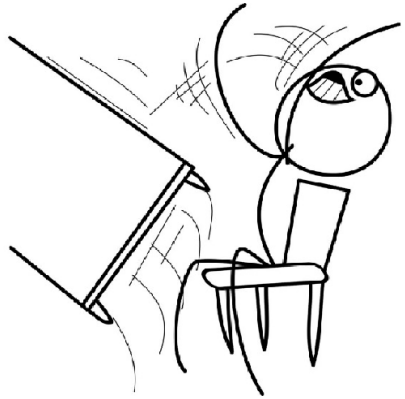
```
65 %let titre_rapport = L'incroyable rapport de Roger ;
66
67 title &titre_rapport. ;
68
69 proc freq data=sashelp.cars ;
70 table make ;
71 quit ;
72
```

Aucun rapport!



Le problème

- Roger sait qu'il est allé trop loin et il veut ajuster le tir.
- Il souhaite que ses collègues sachent que ce n'était qu'une blague.



```
%let titre_rapport = L'"incroyable" rapport de Roger ;

title &titre_rapport. ;

proc freq data=sashelp.cars ;
    table make ;
quit ;
```

Log

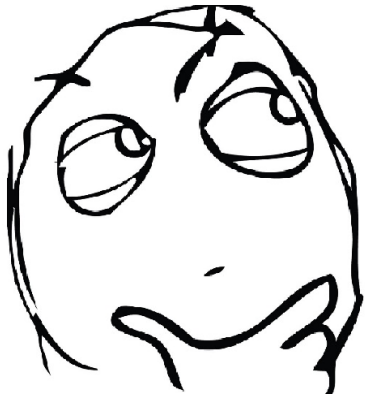
```
65 %let titre_rapport = L'"incroyable" rapport de Roger ;
66
67 title &titre_rapport. ;
68
69 proc freq data=sashelp.cars ;
70 table make ;
71 quit ;
72
```

Toujours aucun rapport!

La racine du problème

- SAS utilise les apostrophes et guillemets comme délimiteurs de chaines de texte.
- SAS s'attends donc à les rencontrer 2 par 2.
- Cette paire délimite une chaine de texte indivisible.
- Pourquoi est-ce que rien ne se passe avec le code SAS de Roger?

SAS attends simplement que Roger termine sas chaine de texte.



La racine du problème

- Retournons au problème original:

```
%let titre_rapport = L'"incroyable" rapport de Roger ;

title &titre_rapport. ;

proc freq data=sashelp.cars ;
    table make ;
quit ;
```

- La chaine de texte a un début mais n’a pas de fin.

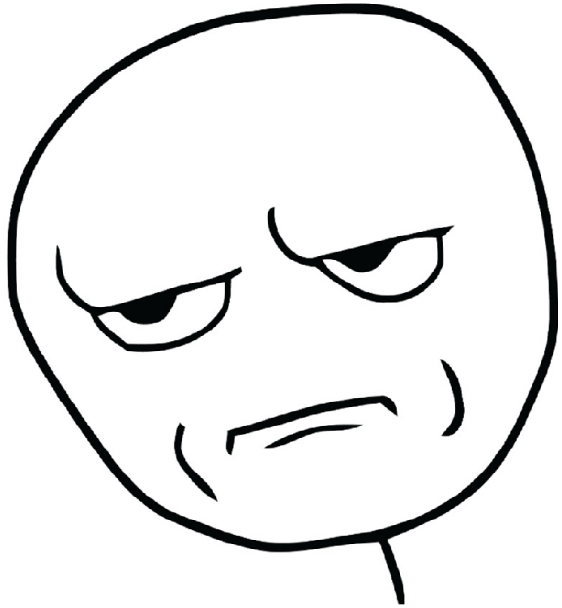
Contourner le problème

- Parfois, contourner le problème est plus rapide.
- Un cas commun est l'utilisation d'apostrophe comme dans « l'incroyable ».
- On peut simplement utiliser un symbole similaire qui ne déclenchera pas une chaine de texte.
- Options:
  - Le véritable apostrophe (')

    désavantage: *ehhh, c'est où sur le clavier ça?*
  - L'accent grave (`)

    désavantage: *Peut avoir un air étrange selon la police utilisée*

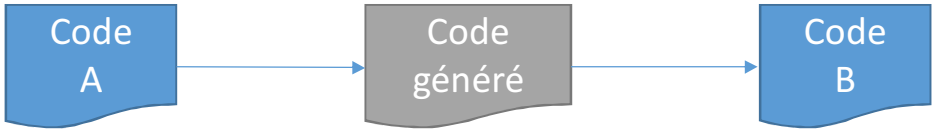
Contourner le problème

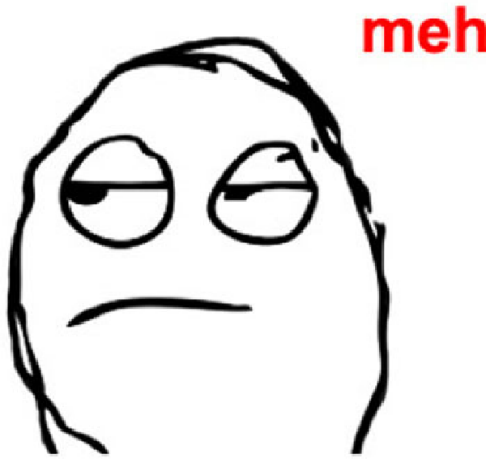


- Ok, ok, les apostrophes ne sont pas toutes utilisées à cette fin.
- Autres cas d'utilisations:
  - Stocker une date entre apostrophes pour une utilisation ultérieure (comme pour une requête pass through)
  - Stocker un emplacement de fichier avec le chemin complet entre apostrophes.
- De toutes façon, c'est votre droit de pouvoir utiliser les apostrophes comme vous le souhaitez.

Contourner le problème

- Dans certains cas, on peut aussi générer du code SAS.
- Code A écrit du code dans un fichier texte (via étape data).
- Ensuite, un code B inclus ce fichier en code SAS (%include).





- Avantage: Fonctionne généralement
- Désavantage: un peu « spaghetti » et difficile à supporter

Faire face au problème

- Pour utiliser les apostrophes, nous devons les masquer.
- %str permet de masquer des éléments lors de la compilation/interprétation d'un code SAS.
- %str permet la résolution de macro variables et programmes.
- %nrstr va plus loin et masque également & and %.
- Truc: Voyez nrstr comme « **N**ot **R**esolving str ».

Faire face au problème

- Plus simplifier, %str and %nrstr sont des moyens de retirer des “pouvoirs” ou habilités aux caractères spéciaux et mots réservés.
- Donc %str et %nrstr sont comme de la kryptonite.
- Ils retirent les habilités des…

“superhéros” suivants:

blank	)	=	NE
;	(	|	LE
¬	+	#	LT
^	--	AND	GE
~	*	OR	GT
, (comma)	/	NOT	
'	<	IN	
"	>	EQ	

Faire face au problème

- Les guillemets et apostrophes sont plus difficiles à masquer.
- Comme un fameux duo dynamique, ils doivent aller en paires, même avec %str and %nrstr.
- Si vous devez en avoir qu'un seul, vous devez le préfixer d'un %.
- Si vous voulez masquer un % alors vous devez mettre %% à l'intérieur de votre fonction %str.
- C'est la seule façon de dire à SAS de ne pas en attendre un second à l'intérieur de la fonction %str ou %nrstr.

Exemple: Pour masquer %', on effectuerait %str(%%%')

Faire face au problème

- Retournons à notre problème:

```
%let titre_rapport = L"incroyable" rapport de Roger ;
title &titre_rapport. ;

proc freq data=sashelp.cars ;
    table make ;
quit ;
```

```
%let titre_rapport = %str(L%'"incroyable" rapport de Roger) ;

title &titre_rapport. ;

proc freq data=sashelp.cars ;
    table make ;
quit ;
```

Faire face au problème

- Problème réglé.
  Simple, n'est-ce pas?
- Pas si vite!
- Analysons le code suivant:

```
%let val = aaa;
%let testval = %str(%'&val%');
%put TRACE: &testval. ;

data _null_;
  val = &testval;
run;
```

Faire face au problème

Code SAS:

```
%let val = aaa;
%let testval = %str(%'&val%');
%put TRACE: &testval. ;

data _null_;
  val = &testval;
run;
```

Log:

```
28
29          %let val = aaa;
30          %let testval = %str(%'&val%');
31          %put TRACE: &testval. ;
TRACE: 'aaa'
32
33          data _null_;
34            val = &testval;
NOTE: Line generated by the macro variable "TESTVAL".
34           'aaa'
             _
             386
              _
              200
ERROR 386-185: Expecting an arithmetic expression.

ERROR 200-322: The symbol is not recognized and will be ignored.

35          run;

NOTE: The SAS System stopped processing this step because of errors.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds
```

Faire face au problème

- Les apostrophes et guillemets perdent leur sens lorsqu'on les masque.
- Si on souhaite leur redonner leur « super pouvoir », il faut les démasquer.
- Il existe une fonction macro qui fait ça.
- Logiquement, elle s'appelle…

  %unquote.

Faire face au problème

- Code SAS corrigé:

```
%let val = aaa;
%let testval = %str(%'&val%');
%put TRACE: &testval. ;

title %unquote(&testval.) ;
proc freq data=sashelp.cars ;
    table make ;
quit ;

data _null_;
  val = %unquote(&testval.);
run;
```

Log:

```
29          %let val = aaa;
30          %let testval = %str(%'&val%');
31          %put TRACE: &testval. ;
TRACE: 'aaa'
32
33          title %unquote(&testval.) ;
34          proc freq data=sashelp.cars ;
35              table make ;
36          quit ;

NOTE: There were 428 observations read from the data set SASHELP.CARS.
NOTE: PROCEDURE FREQ used (Total process time):
      real time           0.01 seconds
      cpu time            0.00 seconds


37
38          data _null_;
39            val = %unquote(&testval.);
40          run;

NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds
```

Quiz!

Code SAS:

```
%let val = aaa;
%let testval = %str(%'&val%');
%put TRACE: &testval. ;

title %unquote(&testval.) ;
proc freq data=sashelp.cars ;
    table make ;
quit ;

data _null_;
  val = %unquote(&testval.);
run;
```

- Quel titre apparaitra dans le rapport de fréquence?

A) aaa

B) 'aaa'

C) Je n'écoutais pas, tu peux reprendre la présentation du début?

Autres façons de masquer et démasquer

- La fonction SUPERQ

  Permet de masquer le contenu d'une variable macro.

  Ex: superq(macvar) masquerait la résolution de la variable macro « macvar »

- Les fonctions BQUOTE et NRBQUOTE

  Elles agissent comme %str et %nrstr sauf que…

  les apostrophes et guillemets peuvent être utilisés sans être en paires!

Allons un peu plus loin

- Amusons nous avec quelques autres exemples…

<u>On veut:</u>

```
%let val = M&M ;

data _null_;
  Bonbon = "&VAL";
run;
```

<u>On utilise:</u>

```
%let val = M%nrstr(&)M ;
```

Ou: `M%str(&)M ;`

```
data _null_;
  Bonbon= "&VAL";
run;
```

Allons un peu plus loin

On veut:

```
%let prenom = Marie-Eve ;
…
/* à l'intérieur d'un pgm macro */
%if &prenom. NE <rien> %then %do ;
        <code SAS impressionnant>
%end ;
…
```

On utilise:

```
%let prenom = Marie%str(-)Eve ;
…
/* à l'intérieur d'un pgm macro */
%if &prenom. NE %str( ) %then %do ;
        <code SAS impressionnant>
%end ;
…
```

Allons un peu plus loin

On veut:

```
%let variable = TST ;
%let hein = '%)*&variable. /* ;

Data test ;
  poubelle = "&hein." ;
Run ;
```

On utilise:

```
%let variable = TST ;
%let hein =
    %str(%'%%%)*&variable. /)%str(*) ;

Data test ;
  poubelle = "&hein." ;
Run ;
```

Bonus: Sommaire exécutif

- Pour utiliser ces super-héros avec des macros, il faut parfois les masquer.
- Masquer quoi que ce soit d'autre ne fait rien.
- %str masque et permet la résolution de macro variables et programmes.
- %nrstr masque mais ne résous pas les macro variables et programmes.
- Pour utiliser un apostrophe ou guillemet seul, il faut le précéder d'un % ou utiliser %bquote et %nrbquote.
- Si vous souhaitez utiliser une chaine délimitée par apostrophes ou guillemets, assurez-vous qu'elle n'est pas masquée (%unquote).

blank	)	=	NE
;	(	|	LE
¬	+	#	LT
^	--	AND	GE
~	*	OR	GT
, (comma)	/	NOT	
'	<	IN	
"	>	EQ	

Références

- Arthur L. Carpenter:

  Quotes within Quotes: When Single (‘) and Double (“) are not Enough
  https://support.sas.com/resources/papers/proceedings15/2221-2015.pdf

  Macro Quoting Functions, Other Special Character Masking Tools, and How To Use Them
  http://www2.sas.com/proceedings/sugi24/Advtutor/p38-24.pdf

- SAS support:

  Macro Quoting: %str and %nrstr functions
  http://support.sas.com/documentation/cdl/en/mcrolref/61885/HTML/default/viewer.htm#a001061290.htm

  Macro Functions: %unquote
  http://support.sas.com/documentation/cdl/en/mcrolref/61885/HTML/default/viewer.htm#a000543618.htm