Manipulating text fields in SAS

Presented by Sata Hackenbruck

6 August 2009

State of Oregon SAS Users Group Meeting

Explore following SAS text functions:

- COMPRESS
- TRANWRD
- SCAN

Here is the problem

I am analyzing pharmacy data and I want to summarize the payments by DRUG GENERIC NAME

However, the current variable that I have looks like this:

drug_generic_name
Oxycodone HCl Tab SR 12HR 10 MG
Oxycodone HCl Tab SR 12HR 20 MG
Oxycodone HCl Tab SR 12HR 40 MG
Oxycodone HCl Tab SR 12HR 80 MG
Gabapentin Tab 300 MG
Gabapentin Cap 400 MG

drug_generic_name
Hydrocodone-Acetaminophen Soln 7.5-500 MG/15ML
Hydrocodone-Acetaminophen Tab 5-500 MG
Fentanyl TD Patch 72HR 50 MCG/HR
Hydrocodone-Acetaminophen Tab 10-650 MG
Fentanyl Citrate Powder
Fentanyl Citrate Inj 0.05 MG/ML

So when I do this

```
proc report data=match;
column drug_generic_name payment;
define drug_generic_name/group 'Generic name';
define payment/analysis sum format=9.0;
title 'Pharmacy Payments by Generic Name';
run;
```

I get something like this

Generic name	Payment
Fentanyl Citrate Buccal Tab 100 MCG (Base Equiv)	668
Fentanyl Citrate Buccal Tab 200 MCG (Base Equiv)	13658
Fentanyl Citrate Buccal Tab 400 MCG (Base Equiv)	20525
Fentanyl Citrate Buccal Tab 600 MCG (Base Equiv)	3853
Fentanyl Citrate Buccal Tab 800 MCG (Base Equiv)	9500
Fentanyl Citrate Inj 0.05 MG/ML	228
Fentanyl Citrate Lollipop 1600 MCG	28473
Fentanyl Citrate Lollipop 200 MCG	11146
Fentanyl Citrate Lollipop 400 MCG	25414
Fentanyl Citrate Lollipop 800 MCG	6931
Fentanyl TD Patch 72HR 100 MCG/HR	123592

continues…

Generic name	Payment
Fentanyl TD Patch 72HR 12 (12.5) MCG/HR	14480
Fentanyl TD Patch 72HR 25 MCG/HR	26953
Fentanyl TD Patch 72HR 50 MCG/HR	75640
Fentanyl TD Patch 72HR 75 MCG/HR	137592
Gabapentin Cap 100 MG	8625
Gabapentin Cap 300 MG	157998
Gabapentin Cap 400 MG	22300
Gabapentin Tab 100 MG	454
Gabapentin Tab 300 MG	97
Gabapentin Tab 400 MG	1819
Gabapentin Tab 600 MG	112974

COMPRESS

COMPRESS(<source>, <chars>, <modifiers>)

```
data match1;
set match;
generic_name=compress(drug_generic_name, ',` ,'D');
```

Modifier 'D‘ removes all numeric characters and
',' **character** removes all the commas.

Other options available to you are:

Modifier ‘A' will remove all alpha characters,
'P' will remove all punctuation,
'S' will remove all spaces
‘K’ will keep all the characters that you specified in <chars> and will remove all the other characters

Example `COMPRESS(var_name,'`','AP')` will remove all alpha characters and punctuation

TRANWRD

TRANWRD(<source>, <target>, <replacement>)

```
data match1;
set match;
generic_name=compress(gpi_generic_name, ',` ,'D');

generic_name=tranwrd(generic_name,"Tab",",");
generic_name=tranwrd(generic_name,"Cap",",");
generic_name=tranwrd(generic_name,"Lollipop",",");
generic_name=tranwrd(generic_name,"Soln",",");
generic_name=tranwrd(generic_name,"Conc",",");
generic_name=tranwrd(generic_name,"TD",",");
generic_name=tranwrd(generic_name,"Inj",",");
generic_name=tranwrd(generic_name,"Buccal",",");
```

SCAN

SCAN(<source> ,<n>, <delimiter(s)>)

<source> specifies any character expression.

<n> specifies a numeric expression that produces the number of the word in the character string you want SCAN to select. The following rules apply:

- If n is negative, SCAN selects the word in the character string starting from the end of the string.
- If |n| is greater than the number of words in the character string, SCAN returns a blank value.

<delimiter> specifies a character expression that produces characters that you want SCAN to use as a word separator in the character string.

Default: If you omit delimiter, SAS uses the following characters:

```
blank . < ( + & ! $ * ) ; ^ - / , % |
```

Tip: If you represent delimiter as a constant, enclose delimiter in quotation marks.

SCAN (continued)

```
data match1;
set match;
generic_name=compress(gpi_generic_name, ',` ,'D');

generic_name=tranwrd(generic_name,"Tab",",");
generic_name=tranwrd(generic_name,"Cap",",");
generic_name=tranwrd(generic_name,"Lollipop",",");
generic_name=tranwrd(generic_name,"Soln",",");
generic_name=tranwrd(generic_name,"Conc",",");
generic_name=tranwrd(generic_name,"TD",",");
generic_name=tranwrd(generic_name,"Inj",",");
generic_name=tranwrd(generic_name,"Buccal",",");

generic_nm=scan(generic_name,1,',');
run;
```

If we run proc report again the result will be:

Generic name	Payment
Fentanyl	378256
Fentanyl Citrate	120396
Gabapentin	361854
Hydrocodone-Acetaminophen	535747
Oxycodone HCl	1775669

Our problem is now solved!

SCAN (continued)

For Example, variable **ArrivalDepartureGates** is as follows:

ArrivalDepartureGates

```
Tokyo, Osaka
Rome, Naples
   …, …
```

```
ArrivalGate = scan(ArrivalDepartureGates,1,',');
```

will return the name of the arrival city

```
DepartureGate = left(scan(ArrivalDepartureGates,2,','));
```

will return the name of the departure city

Other useful functions

You can do a lot of things with SUBSTR and LENGTH functions as well
SUBSTR(<variable>, <position>, <length>)

Here is an example of how to add leading zeroes and standartize the reported National Drug Codes to **11**-digit format:

```
if 7 le (length(service_code)) le 9 and
substr(service_code,1,1) in
   ('0','1','2','3','4','5','6','7','8','9') then
                do while (length(service_code) < 11);
                        service_code=('0'||service_code);
                end;
if length(service_code)=10 then
   if substr(service_code,1,1) = '0' then
           service_code = ('0'||service_code);
      else service_code =
   (substr(service_code,1,5)||'0'||substr(service_code,6,5));
run;
```