

Improve Matches by Using PRXCHANGE to standardize Names

Leonard Landry,
Statistics Canada

Agenda

- Background.
- PRXPARSE and PRXCHANGE.
- Implementation
- Results

Background

- Matching two files by company name
 - Character fields must be exact match
- Some firms had slight name variations
 - Eg. LTD LTD. LIMITED LTEE LIMITEE
- How to standardize these words.
 - PRXPARSE function and call PRXCHANGE Routine

PRXPARSE and PRXCHANGE

- Perl regular expression group of functions and call routines
- Work together to manipulate strings and match patterns
- Enable you make several substitutions in a string in one step.
- Used in a SAS DATA step

PRXPARSE function

- Compiles a Perl regular expression that can be used for pattern matching of a character value
- Syntax
 - *regular-expression-id*=**PRXPARSE** (*perl-regular-expression*)
- Arguments
 - *regular-expression-id* is a numeric pattern identifier that is returned by the PRXPARSE function
 - *perl-regular-expression* specifies a character value that is a Perl regular expression

PRXPARSE function

Example:

```
RX1 = PRXPARSE('s/LIMITED/LTD/');
```

's/LIMITED/LTD/' is a Perl regular expression where:

S specifies the metacharacter for substitution.

LIMITED specifies the regular expression to be searched for.

LTD specifies the replacement value for LIMITED



Call PRXCHANGE routine

- Performs a pattern matching replacement
- Restriction: Use with the PRXPARSE function
- Syntax
 - CALL PRXCHANGE (regular-expression-id, times, old-string)
- Arguments
 - **regular-expression-id** specifies a numeric variable with a value that is a pattern identifier that is returned from the PRXPARSE function.
 - **times** is a numeric constant, variable, or expression that specifies the number of times to search for a match and replace a matching pattern
 - **Old-string** specifies the character expression on which to perform a search and replace



Call PRXCHANGE routine

Example:

```
Call PRXCHANGE(RX1,-1,businessname);
```

RX1 is the regular-expression-id specified in the PRXPARSE function

-1 specifies that all matching patterns are to be replaced

Businessname is the name of the character field containing the character data

Implementation

```
data file1;
  if _N_ = 1 then do;
    retain rx1-rx14; drop rx1-rx14;
    rx1 = prxparse('s/INC\./INC/');
    rx2 = prxparse('s/LIMITED/LTD/');
    rx3 = prxparse('s/LIMITEE/LTD/');
    rx4 = prxparse('s/LTEE/LTD/');
    rx5 = prxparse('s/LTD\./LTD/');
    rx6 = prxparse('s/CORPORATION/CORP/');
    rx7 = prxparse('s/CORP\./CORP/');
    rx8 = prxparse('s/&/ AND /');
    rx9 = prxparse('s/COMPANY/CO/');
    rx10 = prxparse('s/CO\./CO/');
    rx11 = prxparse('s/MANAGEMENT/MGMT/');
    rx12 = prxparse('s/MGMT\./MGMT/');
    rx13 = prxparse('s/,/ /');
    rx14 = prxparse('s/ / /');
  end;
  set mylib.file1;
```



Implementation

```
set mylib.file1;  
name = left(name);  
call prxchange(rx1,-1,name);  
call prxchange(rx2,-1,name);  
call prxchange(rx3,-1,name);  
call prxchange(rx4,-1,name);  
call prxchange(rx5,-1,name);  
call prxchange(rx6,-1,name);  
call prxchange(rx7,-1,name);  
call prxchange(rx8,-1,name);  
call prxchange(rx9,-1,name);  
call prxchange(rx10,-1,name);  
call prxchange(rx11,-1,name);  
call prxchange(rx12,-1,name);  
call prxchange(rx13,-1,name);  
call prxchange(rx14,-1,name);  
run;
```



Implementation

- Standardize names in file1
- Sort File1 by name
- Standardize names in file2
- Sort File2 by name
- Perform the merge by name



Results

- Matches without standardization: 390,394
- Matches with standardization: 443,859
- Improvement: 13.7%
- Results will vary depending on the data.

Questions / Comments



Statistics
Canada

Statistique
Canada

Leonard Landry

Database Manager

Social Analysis Division

R.H. Coats Building, 24th Floor, Section E

Ottawa, Ontario, Canada K1A 0T6

(613) 951-1284 Fax (613) 951-5403

Leonard.landry@statcan.ca

Canada

