DATA Step

How do I remove repeated characters from a text string?

The COMPRESS function has a number of additional arguments which allow for the removal of all instances of certain designated characters, or groups of characters from a text string. In order to remove any example where a character is repeated within a text string requires the additional power of PERL Regular Expressions.

PERL allows for complex validation and substitution rules using metacharacters, character classes and capture buffers.

Consider the code:

```
proc sort data = sashelp.cars (keep  = make)
           out = cars
           nodupkey
    ;
  by make ;
run ;

data cars (drop = pre) ;
  set cars ;
    make  = compress(upcase(make),,'ka') ;
    pre   = prxparse('s/(.)\1+/$1/o') ;
    make2 = prxchange(pre, -1, make) ;
run ;
```

working through the logic, the statement:

make = compress(upcase(make),,'ka') ;

converts the variable into upper case and compresses the result; the 'ka' modifiers K(eep) A(lphanumeric) characters.

The PRXPARSE function parses the Perl Regular expression and stores a numeric identifier. The Regular Expression itself has several parts:

s/

Indicates a substitution instruction and starts the regexp

(.)

The round brackets indicate a Capture Buffer and the period represents any character

\1+

the backslash indicates a metacharacter and the number with the plus sign denoting the

previous 'thing' stored in the Capture Buffer, repeated one or more times

/$1

the forward slash denotes the end of the 'find' and causes the substitution of the 'one or more' characters with the character from Capture Buffer 1

/o

the forward slash denotes the end of the substitution and the o instructs the system to parse the expression just once, as it does not change on subsequent iterations of the DATA step.

The PRXCHANGE function then applies the regexp to the relevant variable - the second argument indicates the number of times to be applied, specifying -1 implies 'every' time the pattern is encountered.

Looking at the outcome:

```
proc print data = cars ;
  where make ne make2 ;
run ;
```

produces the result:

```
Obs    Make        make2

  5    CADILLAC    CADILAC
 12    HUMMER      HUMER
 17    JEEP        JEP
 27    NISSAN      NISAN
 31    SAAB        SAB
```

Unique solution ID: #1012
Author: Alan D Rudland
Last update: 2017-05-12 08:58