LIFECYCLE SOLUTIONS

Minghui Yang, Ph.D

Materials Management Services

Demand Forecast and Planning

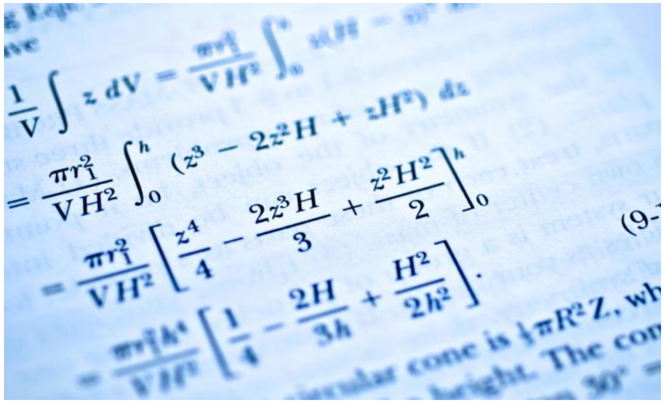
May, 2011

A Clean Way to Convert a Character String by Using SAS Functions

Who We Are?

- Mike and Rob at JAL, 2001

- Support Airline Customers



Our Major Tasks:

- Forecast
- Build math models
- Demand and business planning
- Statistical analysis
- Optimization
- Simulation
- Training

Questions We Faced

In order to merge the Boeing IMM database with data from customers, the key variable part number is used:

- The part numbers in customer data may
  - Not be standardized
  - Need to be converted to the standardized part numbers

- Rules for the conversion
  - (1) all character ‘O’ or ‘o’ will be converted to number ‘0’;
  - (2) all white spaces (blank, TAB etc.) and some special characters such as “*&^%$#@!~,./?+” will be eliminated from the part number;
  - (3) keep the dash ‘-‘ character as is if both sides are numerical characters; otherwise, eliminate it.

Examples of the Conversion Rules

After conversion, the new part numbers should become

OBS	Old part number	**New part number**
1	BACR-XXX1-2-3	BACRXXX1-2-3
2	BACR-XYZ1-2	BACRXYZ1-2
3	BAC1-RXX-2-3	BAC1RXX2-3
4	BACR-RXX-5.3	BACRRXX53
5	BACR-RXX-5.Z	BACRRXX5Z
6	BACR-RXX-5/W	BACRRXX5W
7	BACR-RXX-5 Q	BACRRXX5Q
8	BACO-1RXX-X	BAC0-1RXXX
9	BACO-oXX-X	BAC0-0XXX
10	BACOOO-0XX-X	BAC000-0XXX
11	BOE-0XX-X	B0E0XXX

Challenge of the Task

- In the past, it was done manually by editing an EXCEL file, 5% error rate, and slow. Two people were often involved.
- Lack of examples for solving this type of problems in SAS reference manuals
- C/C++/C# can have the job done. Do we want to call a C/C++/C# subroutine from SAS?
- Need to write a SAS program directly since other people may not understand C/C++/C# codes
- Need to use the new functions in SAS 9.1 or later versions

Algorithm

- Step 1: Using the COMPRESS function to eliminate the white spaces first
- Step 2: Using the LENGTH function to get the length of the part numbers to set-up the maximal length N for the iteration
- Step 3: to find the position of the character ‘O’ or ‘o’ by using the INDEXC function, then using the SUBSTR function and the concatenation operator || to replace them by number ‘0’. This is LOOP 1.
- Step 4: to find the position of all special characters “\./_+?!#%$-*^@~” in LOOP 2; then using SUBSTR function and ANYDIGIT function and concatenation operator || to process those special characters
  - (a) Eliminate all special characters other than “-”
  - (b) A new SAS function ANYDIGIT is used to process the characters on both sides of “-”

My SAS Codes for LOOP 1

```
l=length(partno);
k=0;
part=compress(partno);
j=indexc(part,'Oo');
do while(j>0);
   k=k+1;
   if j=1 then parta=' ';
   else parta=substr(part,1,j-1);
   partb=substr(part,j+1,40-j);
   part=compress(parta||'0'||partb);
   j=indexc(part,'Oo');
end;
```

My SAS Codes for LOOP 2

```
j=indexc(part,'\./_+?!#%$-*^@~');
if j>0 then do;
   parta=' ';
   partb=' ';
   do while(j>0);
      k=k+1;
      parta=compress(parta||substr(part,1,j-1));
      pre=substr(part,j-1,1);
      post=substr(part,j+1,1);
      if anydigit(pre,1)>0 and anydigit(post,1)>0 and j = indexc(part,'-') then parta=compress(parta||'-');
      partb=substr(part,j+1,40-j);
      part=compress(partb);
      j=indexc(part,'\./_+?!#%$-*^@~');
   end;
   part=compress(parta||partb);
end;
ll=length(part);
```

Results

- LOOP 2 is the key
  - (a)Two sequences – PARTA and PARTB
  - (b)PARTA is increasing while PARTB decreasing
  - (c)PARTA=PARTA or PARTA=PARTA||”-”
  - (d)Final PART=PARTA||PARTB

- Accuracy
  - (a)So far 100%
  - (b)Keep some variables for verification/test

- Speed
  - (a) In seconds

BOEING

LOOP 2 Explanations

- Initial PART=BACR-RXX-5.Z
- J=5. Because both sides of “-” are characters, “-” should be eliminated. So now PARTA=BACR and PARTB=RXX-5.Z
- Now PART=RXX-5.Z
- J=4. Because one side of “-” is a character and another side of “-” is digit, “-” should be eliminated. and PARTA=PARTA||’RXX’ i.e. PARTA=BACRRXX. Now PARTB=5.Z
- J=2. Set PART=5.Z so PARTA=BACRRXX5 and PARTB=Z
- Now J=0, iteration ends
- Final PART=PARTA||PARTB
- We have the part BACRRXX5Z
- BACOOO-0XX-X
- Replace O to 0 first (LOOP 1, 3 iterations)
- PARTA=BAC000- (add “-” after PARTA since both sides are digits)
- Now PARTA=BAC000-0XX and PARTB=X
- New PART=PARTA||PARTB=BAC000-0XXX