Paper 2100-2016

Super boost data transpose puzzle

Ahmed Al-Attar, AnA Data Warehousing Consulting LLC, McLean, VA

ABSTRACT

This paper compares different solutions to a data transpose puzzle presented to the SAS® User Group at the US Census Bureau (CenSAS). The presented solutions ranged from a SAS® 101 multi-step solution to an advanced solution utilizing not widely known techniques yielding **85%** run time savings!

INTRODUCTION

While working with the International Data Base (IDB) data at the US Census Bureau, I needed to find a way to transpose a wide horizontal table with 101 numeric columns and two rows per group, into a thinner table with 101 rows per group and new column representing the aggregation of two columns per row.

scheme	regno	CTY	YR	SEX	MAXAGE	P000	P001	P002	P003	P004	….	….	….	P100
1	1	AA	1981	2	100	553	559	540	516	493				X
1	1	AA	1981	3	100	529	536	517	494	472				X
1	1	AA	1982	2	97	549	551	558	539	516				X
1	1	AA	1982	3	100	561	528	536	517	493				X
1	1	AA	1983	2	97	569	547	550	559	539				X
1	1	AA	1983	3	100	582	559	528	536	517				X
1	1	AA	1984	2	98	585	566	547	549	559				X
1	1	AA	1984	3	100	600	580	559	527	536				X
1	1	AA	1985	2	99	587	577	562	542	545				X
1	1	AA	1985	3	100	603	598	578	557	526				X
1	1	AA	1986	2	100	577	580	571	556	536				X

Figure 1: Original IDB Birth Rate table

Original data characteristics:

- ***Sorted By***: **Scheme, regno, cty, yr, sex**
- There are 101 age variables (P000 – P100)
- Each Combination of **Scheme, regno, cty, yr** has a row for **Male (Sex=2)** and a row for **Female (Sex=3)**
- **240,616** observations

Resulting data table should have

- **Male** & **Female** Records transposed
- **Both** Column is calculated by summing Male & Female numbers
- **popAge** is 0-100
- 12+ Millions observation ((240,616/2) * 101)

scheme	regno	CTY	YR	Popage	male	female	both
1	1	AA	1981	0	553	529	1082
1	1	AA	1981	1	559	536	1095
1	1	AA	1981	2	540	517	1057
1	1	AA	1981	3	516	494	1010
1	1	AA	1981	4	493	472	965
1	1	AA	1981	5	477	457	934
1	1	AA	1981	6	471	450	921
1	1	AA	1981	7	472	452	924
1	1	AA	1981	8	483	463	946
1	1	AA	1981	9	499	476	975
1	1	AA	1981	10	514	490	1004
1	1	AA	1981	11	532	508	1040
1	1	AA	1981	12	556	530	1086
1	1	AA	1981	13	583	558	1141
1	1	AA	1981	14	613	588	1201
1	1	AA	1981	15	644	622	1266
1	1	AA	1981	16	672	652	1324
1	1	AA	1981	17	689	670	1359
1	1	AA	1981	18	694	670	1364
1	1	AA	1981	19	686	657	1343
1	1	AA	1981	20	676	639	1315
1	1	AA	1981	21	664	623	1287
1	1	AA	1981	22	649	610	1259
1	1	AA	1981	23	628	606	1234

Figure 2: Desired Transposed Birth Rate table

It was very important to come up with a solution that has the least amount of data reads and the fastest run time.

Once I developed my solution, I wanted to validate it and see if there could a better solution. That's when I reached out to the SAS® users group at the US Census Bureau (CenSAS), and posted it as a Data Transpose Puzzle. I got two helping solutions with different level of complexity, and approach.

SOLUTION 1

This solution was contributed by a Branch Chief at the Census. It consisted of 3 steps (Data Step, Proc Sort, Data Step).

```
%let g_srcDsName = censas.idb194;

 data b (keep = scheme regno cty yr popage pop);
   set &g_srcDsName;
   array popagea(101) 8 p000-p100;
   do i = 1 to 101;
      popage = i - 1;
      if sex = '2' then pop = popagea{i};
      else if sex = '3' then pop = popagea{i};
      output;
   end;
run;

proc sort data=b;
   by scheme regno cty yr popage;
run;

data c (drop = pop);
   set b;
   by scheme regno cty yr popage;
   retain male female;
   if first.popage then male = pop;
   if last.popage then
   do;
      female = pop;
      both = male + female;
      output;
   end;
run;
```

Pros:

- Clear & Simple
- Easy to follow
- Adoptable by levels of skills

Cons:

- Ignored data characteristics (Sorted by)
- Multiple data reads
- Intermediate table
- Typical SAS® 101 approach
- Suitable for teaching but not for Production deployment

SOLUTION 2

This solution was contributed by a SAS® on-site Technical Advisor at the Census Bureau. It consisted of 2 steps (Proc Transpose, Proc SQL).

```
%let g_srcDsName = censas.idb194;


Proc transpose data=&g_srcDsName( drop=maxage sex)
  out=WORK.alattartest2( rename= (sex_1=Male
sex_2=Female))
  name=Popagetem
  prefix=sex_;
  by scheme regno CTY YR;
run;


Proc sql;
  create table final as
  select   scheme ,regno ,CTY ,YR
    ,substr(Popagetem,2,4) as Popage
    ,Male ,Female ,sum(Male+Female) as both
  from alattartest2;
quit;
```

Pros:

- Utilized data characteristics (Sorted by)
- Easy to follow
- Mixing Procs & SQL

Cons:

- Multiple data reads
- Proc Transpose performs multiple internal data reads
- Intermediate table

SOLUTION 3

This is the solution I had developed prior to ask for input from the SAS® users group. It consisted of a single step (Data Step).

```
%let g_srcDsName = censas.idb194;


DATA WORK.idb194_transposed (KEEP=scheme regno cty yr Popage male female both );
  ARRAY p P000-P100;
  ARRAY MP MP000-MP100;
  ARRAY FP FP000-FP100;
  LENGTH   pFirstPos $20 init_rb8_str $808;
  RETAIN MP: FP: init_rb8_str;
  if (_n_=1) then
  do;
    pFirstPos = ADDRLONG(p000);
    init_rb8_str = PEEKCLONG(ADDRLONG(p000));
  end;

  SET &g_srcDsName;
  BY   scheme regno cty yr;

  LENGTH   Popage 4 male female both 8;
  LABEL   PopAge = 'Population at Age'
            male   = 'Male Population'
            female = 'Female Population'
            both   = 'Both sexes Population';

  /* Find the destination address */
  if (sex=2) then   call POKELONG(PEEKCLONG(ADDRLONG(p000)),ADDRLONG(mp000),808);
  else   call POKELONG(PEEKCLONG(ADDRLONG(p000)),ADDRLONG(fp000),808);

  IF (last.yr) THEN
  DO i=1 to dim(p);
    popAge = i-1;
    male   = mp[i];
    female = fp[i];
    both   = Sum(male,female);
    OUTPUT;
  END;
RUN;
```

Pros:

- Utilized data characteristics (Sorted by)
- Single data read
- Single Output – No Intermediate table(s)

Cons:

- Uses unfamiliar yet very powerful functions
- Requires advanced skills level

RUN TIMES COMPARISON

In order to have a fair and comprehensive comparison, I used the same two data sets with all three solutions on the same SAS® platform. The table below illustrates the differences in their run times.

Table Sizes	Solution 1 3 Steps mm:ss.ss	Solution 2 2 Steps mm:ss.ss		Solution 3 1 Step mm:ss.ss		
	run-time	run-time	% of S1	run-time	% of S1	% of S2
Sample Data (2,310 records)	00:00.36	00:00.26	**27.78%**	**00:00:08**	**77.78%**	**69.23%**
Full Data (240,616 records)	00:28.71	00:23.80	**17.10%**	**00:03.42**	**88.08%**	**85.63%**

Table 1: Solutions run times and percent of improvement

CONCLUSION

Some of the unwieldy known features of the SAS® programming language can offer more elegant and efficient solutions to data manipulation problems.

Whether you are a SAS® programmer with many years of experience or a novice user who is responsible for maintaining legacy programs, implementing updated approaches can allow you to streamline your SAS® applications, expedite the development and debugging process, and minimize future maintenance of the code.

Investigating and researching alternative approaches and solutions is a worthwhile investment for any SAS® programmer, regardless of their level of experience.

REFERENCES

Paul M. Dorfman, and Lessia S. Shajenko 2011. "From Obscurity to Utility: ADDR, PEEK, POKE as DATA Step Programming Tools". Proceedings of the SouthEast SAS Users Group 2011, http://analytics.ncsu.edu/sesug/2011/BB07.Dorfman.pdf

Paul M. Dorfman, and William W. Viergever 2012. "Straight from Memory: ADDR, PEEK, POKE as SAS® Programming Tools". Proceedings of SAS Global Forum 2012, http://support.sas.com/resources/papers/proceedings12/223-2012.pdf

Ed Heaton 2009, "Resetting Variables to Zero". Proceedings of the SouthEast SAS Users Group 2009, http://analytics.ncsu.edu/sesug/2009/SC011.Heaton.pdf

Carol A. Martell 2013. "PEEKing at Roadway Segments". Proceedings of the SouthEast SAS Users Group 2013, http://analytics.ncsu.edu/sesug/2013/BtB-07.pdf

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Ahmed Al-Attar
AnA Data Warehousing Consulting, LLC.
McLean, VA 22101
Cell Phone: 703-477-7972
E-mail: ahmed.al-attar@anadwc.com
Web: www.anadwc.com

TRADEMARKS