Paper 106-2011

# Assembling Data Set for Use in Geographic Information System (GIS): Use of PROC TRANSPOSE in Macro

Saiful Momen, University of Hawaii at Manoa, Honolulu, HI

## ABSTRACT

With the increasing popularity of Geographic Information System (GIS), outputs of data analyses are often mapped, and analyzed for spatial patterns. The format of GIS files requires that the data set to be joined with map features contain observations (i.e. rows) corresponding to each spatial feature (e.g. a county, or a hospital). Often published data sets, particularly the ones that come out at regular intervals in the same tabular format (e.g. employment data by county) do not have rows representing the spatial features. This paper shows a simplified case of a project that used BASE SAS® to assemble a "GIS-ready" data set out of multiple data tables that were published over the years on sector-wise (i) employment and (ii) gross output, and disaggregated by districts in Bangladesh. Source files were in multiple MS Excel files with multiple worksheets. Use of CALL SYMPUT and macro variables make the code flexible with the number of input tables, and the number of variables and rows in them.

## 1. INTRODUCTION: BASIC STRUCTURE OF (VECTOR) GIS

Geographic information system uses computers and software to link location of events and objects to attributes of those events and objects. The events and objects are represented by what is called features, which can be either a point, or a line, or an area. For example, a street is often represented as a line, a bus stop as a point, or a land use zone as an area. Thus, a representation of a set of objects or events on earth becomes a set of features that is location-specific, i.e. spatially-referenced. Attached with each feature in GIS is description (attributes) of the particular object that the feature represents. For example, we can have a *map* of all counties in the Mid-West, and a *table* linked with it that contains information of interest (i.e. variables) on each county (e.g. county population, population growth rates over the last decade, number of jobs gained over the last year etc.).

The original project from which the simplified demonstration of this paper comes assembled published statistical tables from three different years, for 64 administrative districts, and had dozens of variables. For the sake of simplicity, we explain the code with four counties of the state of Hawaii, and that too with only two hypothetical sectors, two attributes of the sectors, and two years of imaginary data (2000 and 2005).

In Geographic Information System (GIS), the hypothetical situation will be represented by the following figure and associated table with data on:

1. Two sectors of the economy (Construction and Hotel),
2. Two indicators of each sector: (i) Number of jobs and (ii) Gross output.

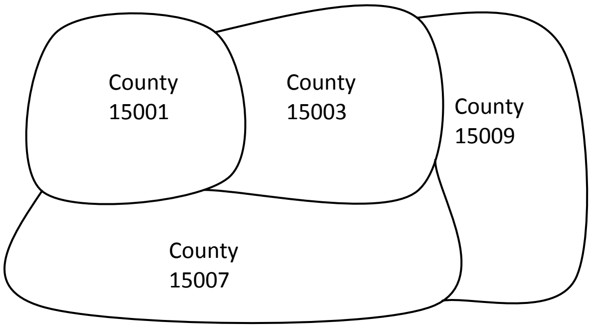

Figure 01: Schematic representation of four spatial entities (areas, as opposed to line or point, in this case)

Table 01: Structure of an attribute table in GIS (This one corresponds to Fig. 01)

| County Id | Yr 2000 # of Construction Jobs | Yr 2000 # of Hotel Jobs | Yr 2005 # of Construction Jobs | Yr 2005 # of Hotel Jobs | Yr 2000 # of Construction Gross Output | Yr 2000 # of Construction Gross Output | Yr 2000 # of Construction Gross Output | Yr 2000 # of Construction Gross Output |
|---|---|---|---|---|---|---|---|---|
| 15001 | | | | | | | | |
| 15003 | | | | | | | | |
| 15007 | | | | | | | | |
| 15009 | | | | | | | | |

The beauty of Geographic Information System (GIS) is that by linking spatial and attribute information, it allows us to prepare many visual and analytical tools. One common example is thematic mapping based on attribute information (e.g. different shades of color for counties with different job gain numbers). Another example is use of spatial statistics to investigate spatial patterns (e.g. level of clusteredness of counties with job gain).

## 2. PREPARING A SERIES OF TABLES FOR GIS

Official statistics on many aspects of economy, or society are published often at regular intervals for different geographic units. After several rounds of publication, we have at hand tabular data for different points in time (hypothetical example: 1995, 2000, 2005 data sets on the number of jobs in different industries for all the counties of the U.S.).

To make these tables compatible with the structure of GIS, they need to be assembled in a tabular form where rows represent each geographic unit and the variables are constructed from segments of (1) file name, (2) column title, and (3) row title.

In this paper we show a simplified case with only two sectors of employment (hotels and construction) for each of four counties of the state of Hawaii for the years 2000 and 2005. The eight (4 counties * 2 years of data) tables will look like the one below.

Table 02: Sector-wise number of jobs and gross output, Hawaii County (code 15001), 2000
(Microsoft Excel worksheet name y00C15001)

| Sector **(s)** | Jobs (**Jb**) | Gross Output (**Go)** |
|---|---|---|
| Hotels **(H)** | | |
| Construction **(C)** | | |

As described in the basic structure of GIS, the data in the 8 tables need to be arranged as attributes of counties. In other words, our desirable table will be like this (the arrows show how existing filename, row titles, and variables names are parsed and put together to assemble the observations and the variables of the desired data set):

**Table 03: The Required Format of the Data Set that We have to Assemble to join/relate with GIS file**

| County Code<br>15001 = Hawaii<br>15003= Honolulu<br>15007= Kauai<br>15009= Maui | Number of jobs in **H**otels in 2000<br>**y00HJb** | Number of jobs in **H**otels in 2005<br>**y05HJb** | Number of jobs in **C**onstruction in 2000<br>**Y00CJb** | Number of jobs in **C**onstruction in 2005<br>**y05CJb** | Gross Output, **H**otels, 2000<br>**Y00HGo** | Gross Output, **H**otels, 2005<br>**y05HGo** | Gross Output, **C**onstruction 2000<br>**Y00CGo** | Gross Output, **C**onstruction 2005<br>**y05CGo** |
|---|---|---|---|---|---|---|---|---|
| 15001 | 50,000 | 51,000 | 30,000 | 40,000 | $200 million | $220 million | $150 million | $210 million |
| 15003 | | | | | | | | |
| 15007 | | | | | | | | |
| 15009 | | | | | | | | |

**Table 04: Files We have from Published Sources**

Microsoft Excel worksheet name: y00C15001 (Hawaii County)

| Sector **(s)** | Jobs2000 (**Jb**) | Gross Output, 2000 (**Go**) |
|---|---|---|
| Hotels **(H)** | 50,000 | $200 million |
| Construction **(C)** | 30,000 | $150 million |

We have published data in the same format for 3 other counties. All data Hypothetical.

Microsoft Excel worksheet name: y05C15001 (Hawaii County)

| Sector **(s)** | Jobs 2005 (**Jb**) | Gross Output, 2005 (**Go**) |
|---|---|---|
| Hotels **(H)** | 51,000 | $220 million |
| Construction **(C)** | 40,000 | $210 million |

## 3. MICROSOFT EXCEL TABLES FOR THE SAS CODE

In the process of running the code below, SAS will create new variables that will result from concatenating parts or whole of existing variables, filenames and value of first variable. One note of caution is that Microsoft Excel allows special characters in column heading. But SAS does not allow them, nor does it allow a variable name to start with number. As the first step, the column and row titles in Microsoft Excel files have to be modified so they are named consistently. To avoid surprises, it is a good idea to delete any texts other than the data and the modified row and column titles. Following code is written based on the rows and variables in Table 04 that shows structure of published tables.

## 4. THE STEPS IN THE SAS CODE

The SAS code needs to perform following steps.

1. Import the Microsoft Excel worksheets as SAS data sets.
2. Put year (e.g. y00) as a prefix to all variable names.
3. In each SAS data set, add a variable “county” to denote the county the data set belongs to. The variable will have the same value for all observations. This variable will be the BY variable in the PROC TRANSPOSE step.
4. Rearrange the SAS data sets into one line using PROC TRANSPOSE with county as the BY variable.
5. Append the lines of data (each representing one county now) to generate a data set for all counties for a particular year.
6. MERGE the yearly data sets into the final data set to be exported as a dbf file. This dbf file will be joined with the GIS map of counties.

The intermediate outputs at the end of each step will be as follows.

### STEP 01: IMPORT THE MICROSOFT EXCEL WORKSHEETS AS SAS DATA SETS

Output at the end of this step: Eight SAS data sets (4 counties for each of years 2000 and 2005)

| Sector **(s)** | Jobs 20** (**Jb**) | Gross Output, 20** (**Go**) |
|---|---|---|
| Hotels **(H)** | | |
| Construction **(C)** | | |

### STEP 02: PUT YEAR (E.G. Y00) AS A PREFIX TO ALL VARIABLE NAMES

Output at the end of this step: Eight SAS data sets in following format (shown below with year 2000)

| Sector **(s)** | Yr00Jb | Yr00Go |
|---|---|---|
| Hotels **(H)** | | |
| Construction **(C)** | | |

### STEP 03: IN EACH SAS DATA SET, ADD A VARIABLE “COUNTY” TO DENOTE THE COUNTY THE DATA SET BELONGS TO

Output at the end of the step: Eight data sets (shown with year 2000, and County Hawaii, code 15001):

| Sector **(s)** | Yr00Jb | Yr00Go | County |
|---|---|---|---|
| Hotels **(H)** | | | 15001 |
| Construction **(C)** | | | 15001 |

### STEP 04: REARRANGE THE SAS DATA SETS INTO ONE LINE USING PROC TRANSPOSE WITH COUNTY AS THE BY VARIABLE

Output at the end of the step: Eight tables (Four counties * 2 years, shown here with county 15001 and Yr. 2000)

| County | Y00HJb | Y00HGo | Y00CJb | Y00CGo |
|---|---|---|---|---|
| 15001 | | | | |

### STEP 05: APPEND THE LINES OF DATA (EACH REPRESENTING ONE COUNTY NOW) TO GENERATE A DATA SET FOR ALL COUNTIES FOR A PARTICULAR YEAR

Output at the end of the step: Two tables, one for each year (shown here with yr. 2000), in the following format

| County | Y00HJb | Y00HGo | Y00CJb | Y00CGo |
|---|---|---|---|---|
| 15001 | | | | |
| 15003 | | | | |
| 15007 | | | | |
| 15009 | | | | |

### STEP 06: MERGE THE YEARLY DATA SETS INTO THE FINAL DATA SET TO BE EXPORTED AS A DBF OR EXCEL FILE.

Output at the end of this step: One (final) table

| County | Y00HJb | Y00HGo | Y00CJb | Y00CGo | Y05HJb | Y05HGo | Y05CJb | Y05CGo |
|---|---|---|---|---|---|---|---|---|
| 15001 | | | | | | | | |
| 15003 | | | | | | | | |
| 15007 | | | | | | | | |
| 15009 | | | | | | | | |

## 5. THE CODE

Please note that there are six different codes below that were written for each of the above steps. It is possible to write a much shorter code to achieve what six steps below do by running sequentially. However efficiency was sacrificed to illustrate the conceptual organization of the project. Even though the simplified case has only 2 years of data on only two sectors and two variables (“Job” and “Gross Output”), the code with minor modification can accommodate more input tables and more rows and variables.

### STEP 01: IMPORT THE MICROSOFT EXCEL WORKSHEETS AS SAS DATA SETS

Output at the end of this step: Eight SAS data sets (4 counties for two years, 2000 and 2005)

| Sector **(s)** | Jobs 20** (**Jb**) | Gross Output, 20** (**Go**) |
|---|---|---|
| Hotels **(H)** | | |
| Construction **(C)** | | |

```
%MACRO import(book,list);
%LET num=1;
%LET sheet_name=%SCAN(&list.,&num.);
%PUT file &num = &sheet_name.;
   %do %while(&sheet_name. ne ) ;
     PROC IMPORT DATAFILE= "Z:\SASstuffs\experim\&book..xls"
         OUT = &sheet_name. DBMS=excel2000 REPLACE;
        GETNAMES=yes;
        SHEET="&sheet_name";
     RUN;
        %if %sysfunc (exist(&sheet_name.)) %then %do;
           DATA &sheet_name;
           SET &sheet_name;
           Run;
        %end;
     %LET num=%EVAL(&num.+1);
     %LET sheet_name=%SCAN(&list,&num.);
   %end;
%mend import;
```

```
%macro ImportNow(yr);
   %LET wkbk = y&yr.;/* Files y00.xls and y05.xls have the worksheets for each
county for years 2000 and 2005 respectively*/
   %do i = 15001 %to 15009;
      %let files = y&yr.c&i.;
      %Import(&wkbk.,&files.);
      %Import(&wkbk.,&files.);
   %end;
%mend;
%ImportNow(00);
%ImportNow(05);
```

### STEP 02: PUT YEAR (e.g. Y00) AS A PREFIX TO ALL VARIABLE NAMES

Output at the end of this step: Eight SAS data sets in following format (shown below with year 2000)

| Sector **(s)** | Yr00Jb | Yr00Go |
|---|---|---|
| Hotels **(H)** | | |
| Construction **(C)** | | |

```
%macro AppendtoVarName(yr=);
   %do i = 15001 %to 15009;
      %let member = y&yr.C&i;
      DATA _null_;
      SET sashelp.vcolumn;
      where memname=upcase("&member")
      ;
      CALL SYMPUT("name"||trim(left(put(_n_,best.))),trim(left (name)));
      CALL SYMPUT("Count",trim(left(put(_n_,best.))));
      run;

         %if %sysfunc (exist(y&yr.c&i)) %then %do;
         DATA y&yr.c&i.a ;
         SET y&yr.c&i ;
            %do j = 2 %to &count; /*No need to add year as prefix, i.e. y00 or y05,
to variable "sector", which is the first variable in the dataset */
               rename &&name&j = y&yr&&name&j.;
            %end;
         %end;
   %end;
   run;
%mend AppendtoVarName;
%AppendtoVarName(yr=00 );
%AppendtoVarName(yr=05 );
```

### STEP 03: IN EACH SAS DATA SET, ADD A VARIABLE “COUNTY” TO DENOTE THE COUNTY THE DATA SET BELONGS TO

Output at the end of the step: Eight data sets (shown with year 2000, and County Hawaii, code 15001):

| Sector **(s)** | Yr00Jb | Yr00Go | County |
|---|---|---|---|
| Hotels **(H)** | | | 15001 |
| Construction **(C)** | | | 15001 |

```
%macro putCounty(yr);
   %do i = 15001 %to 15009;
      %if %sysfunc (exist(y&yr.c&i.a)) %then %do;
         DATA y&yr.C&i.x (label = County &i Year &yr with variable County);
         SET y&yr.C&i.a;
         County = &i;
         run;
      %end;
   %end;
```

```
%mend putCounty;

%putCounty (00);
%putCounty (05);
```

## STEP 04: REARRANGE THE SAS DATA SETS INTO ONE LINE USING PROC TRANSPOSE WITH COUNTY AS THE BY VARIABLE

Output at the end of the step: Eight datasets (shown with year 2000, and county 15001)

| County | Y00HJb | Y00HGo | Y00CJb | Y00CGo |
|---|---|---|---|---|
| 15001 | | | | |

```
%macro transpose(yr);
   %do i = 15001 %to 15009;
      %let member = y&yr.C&i.x;
      DATA _null_;
      SET sashelp.vcolumn;
      where memname=upcase("&member")
      ;
      CALL SYMPUT("name"||trim(left(put(_n_,best.))),trim(left (name)));
      CALL SYMPUT("Count",trim(left(put(_n_,best.))));
      run;
         %do j = 2 %to &count-1; /*the last variable is county*/
            %if %sysfunc (exist(y&yr.c&i.x)) %then %do;
               PROC TRANSPOSE data= y&yr.C&i.x out = y&yr.c&i.&&name&j. prefix =
&&name&j.;
               by County;
               id s;
               var &&name&j.;
               run;
            %end;
         %end;
   %end;
   %do k = 15001 %to 15009;
      DATA y&yr.C&k.z (label = Year &yr. County &k. data in a single line);
      MERGE
      %do j = 2 %to &count-1;
         y&yr.c&k.&&name&j.
      %end;
      ;
      by County;
      run;
   %end;
%mend transpose;
%transpose(00);
%transpose(05);
```

## STEP 05: APPEND THE LINES OF DATA (EACH REPRESENTING ONE COUNTY NOW) TO GENERATE A DATA SET FOR ALL COUNTIES FOR A PARTICULAR YEAR

Output at the end of the step: Two tables, one for each year (shown here for yr. 2000)

| County | Y00HJb | Y00HGo | Y00CJb | Y00CGo |
|---|---|---|---|---|
| 15001 | | | | |
| 15003 | | | | |
| 15007 | | | | |
| 15009 | | | | |

```
%Macro append(yr);
   DATA y&yr.;
   SET
      %do i = 15001 %to 15009;
         y&yr.C&i.z
      %end;
   ;
   run;
%mend;
%append(00);
%append(05);
```

### STEP 06: MERGE THE YEARLY DATA SETS INTO THE FINAL DATA SET TO BE EXPORTED AS A DBF OR EXCEL FILE

Output at the end of this step:

| County | Y00HJb | Y00HGo | Y00CJb | Y00CGo | Y05HJb | Y05HGo | Y05CJb | Y05CGo |
|---|---|---|---|---|---|---|---|---|
| 15001 | | | | | | | | |
| 15003 | | | | | | | | |
| 15007 | | | | | | | | |
| 15009 | | | | | | | | |

```
   DATA final;
      MERGE y00 y05;
      by county;
   run;
```

### STEP 07: EXPORT IN DBF FORMAT

```
   PROC EXPORT
      data=Final
      outfile="Z:\...\final.dbf"
      DBMS=DBF REPLACE;
   run;
```

## 6. ADDITIONAL COMPLICATING FACTOR NOT COVERED IN THIS SIMPLIFIED EXAMPLE

Sectors in economy in many countries are subject of elaborate classification system. Activities will be classified with progressively higher number of digits where digits on the right represent a subcategory of the digit(s) on the left. For example, in the most recent North American Industry Classification System (NAICS), 31 is manufacturing, 314 is textile product mills, and 3141 is textile furnishing mills.

If the employment data is reported at 4 digit level in the excel files, and we want aggregation at major industry level (2-digit), we have to (i) %SUBSTR the first two digits of the sector code, and (ii) use PROC MEANS with BY and Sum to generate total employment for major industry categories, before we can TRANSPOSE the data sets.

## 7. CONCLUSION

PROC TRANSPOSE is a relatively less frequently used procedure. However, for assembly of data for GIS, it comes very handy in a number of situations. Use of CALL SYMPUT and SAS Macro helps to flexible and versatile codes that process the input data files even when the number of rows or variables change from one file to another (from year to year in our example).

## CONTACT INFORMATION

Saiful Momen
Department of Urban and Regional Planning
Univerisyt of Hawaii at Manoa
2424 Maile Way, Rm 107
Honolulu, HI 96822
Phone: 808-956-8163
Email: momen@hawaii.edu