Introduction to SAS

Hsueh-Sheng Wu

Center for Family and Demographic Research

November 1, 2010

Outline

- What is SAS?
- Things you need to know before using SAS
- SAS user interface
- Using SAS to manage data
- Strengths and limitations of SAS
- Conclusions

What Is SAS?

- SAS is one of statistical software packages, just like Stata or SPSS.
- What does SAS do?
  - Data management
  - Data analysis
  - Ability to use graphs to present analysis results
- How does SAS differ from other statistical software?
  - Different user interface
  - Different data format
  - Different syntax rules
  - Can handle large data files
  - Can work on multiple data sets simultaneously

Things You Need to Know Before Using SAS

- SAS reads one observation at a time.
- There are two steps in using SAS to analyze data. The DATA step is for data management and the PROC step is for statistical analysis.
- The default setting of SAS is to construct and analyze data in a working directory. The content of this directory will be deleted when SAS is closed. Thus, if you want to keep your work permanently, you need to save it in another directory.
- Numeric variables are different from string variables.

Things You Need to Know Before Using SAS (Cont.)

- Make references to directories, files, variables
  - _NUMERIC_ specifies all numeric variables that are currently defined in the DATA step.
  - _CHARACTER_ specifies all string variables that are currently defined in the DATA step.
  - _ALL_ specifies all the numeric and string variables that are currently defined in the DATA step.
  - The colon ":" represents a wild card. For example, you can use "m:" to refer to two variables, make and mpg, simultaneously.
  - Two dashes "--" represents "from … to". Thus, "make—mpg" represents make and mpg and every variables between them.
  - The statement "Variable A -numeric- variable B" represents all numeric variables from Variables A to B.
  - The statement "Variable A -character- Variable B" represents all the string variables from variables A and B.

Things You Need to Know Before Using SAS (Cont.)

- Know the differences among variables, variable values, variable format.

- Some important rules of writing SAS syntax
  - A SAS statement can start in any column.
  - A SAS statement can be in upper- or lower-case.
  - A SS statement can be on the same line as other statements.
  - A Ss statement can continue on the next line (as long as you don't split a word in two)
  - Every SAS statement ends with a semicolon.

Things You Need to Know Before Using SAS (Cont.)

- Rules of naming a variable
  - Names must start with a letter or an underscore (_).
  - Names can contain only letters, numerals, or underscores (_), but not other special characters, such as %, $, !, *, &, #, or @.
  - Names must be 32 characters or fewer in length.
  - Letters can be uppercase, lowercase, or a mixture of the case
  - The above rules applied to naming format
  - Similar rules applied to naming other things in SAS (e.g., informat, libref, and fileref), except that the length of name is 31 characters for a informat, 8 charcters for a libref and a fileref must be 8 characters or fewer in length, and member names for versioned data sets must be 28 characters or fewer.
  - Don't use the following names: _N_, _ERROR_, _CHARACTER_, _NUMERIC_, _ALL_, SASHELP, SASMSG, SASUSER, WORK,_NULL_, _DATA_, _LAST_, SASCAT, and SYS. SAS already uses these names for special purposes.

- Missing values is treated as the *smallest* possible value.

- Two ways of Adding comments in SAS command files
  - * this is a comment ;
  - /*
    This is a comment
    */

SAS User Interface

- Three main windows
  - Explorer window for looking at the data
  - Editor window for writing a SAS command file
  - Log window for errors in the SAS program
- An additional window – the output window
  - The output window automatically pops up after you execute a SAS command file that produces SAS outputs
- The steps of using SAS
  - Using Editor window to write a SAS command file
  - Execute the command file
  - Check if there are the error messages in the log window
  - Check the output in the output window

Using SAS to Manage Data

- What are data?
- Use SAS to read in data
- Take a look at the data file
- Change the order of observations or variables
- Modify variables
- Add labels
- Create new variables
- Merge data
- Create a subset of data

What Are Data?

Raw data:

```
AMC Concord      40992232.5112930186401213.579999923706050
AMC Pacer         4749173  3113350173402582.529999971389770
AMC Spirit       379922   3122640168351213.079999923706050
Buick Century     48162034.516325019640196 2.93000006675720
Buick Electra     7827154  4204080222433502.410000085830690
.
.
.
VW Dasher         71402342.512216017236 973.740000009536741
VW Diesel         5397415  315204015535 903.779999971389771
VW Rabbit         4697254  315193015535 893.779999971389771
VW Scirocco       6850254  216199015636 973.779999971389771
Volvo 260        119951752.5143170193371632.980000019073491
```

Final data:

# of observation	make	price	mpg	rep78	headroom	trunk	weight	length	turn	displacement	gear_ratio	foreign
1	AMC Concord	4099	22	3	2.5	11	2930	186	40	121	3.58	Domestic
2	AMC Pacer	4749	17	3	3	11	3350	173	40	258	2.53	Domestic
3	AMC Spirit	3799	22		3	12	2640	168	35	121	3.08	Domestic
4	Buick Century	4816	20	3	4.5	16	3250	196	40	196	2.93	Domestic
5	Buick Electra	7827	15	4	4	20	4080	222	43	350	2.41	Domestic
.	.											
.	.											
.	.											
70	VW Dasher	7140	23	4	2.5	12	2160	172	36	97	3.74	Foreign
71	VW Diesel	5397	41	5	3	15	2040	155	35	90	3.78	Foreign
72	VW Rabbit	4697	25	4	3	15	1930	155	35	89	3.78	Foreign
73	VW Scirocco	6850	25	4	2	16	1990	156	36	97	3.78	Foreign
74	Volvo 260	11995	17	5	2.5	14	3170	193	37	163	2.98	Foreign

What Are Data? (Continued)

- The final data set looks just like an Excel table.

- Each column represent a variable, except the first column that I added to indicate the number of observations in the data.

- Each row represents a observation, except the first row that I added to indicate the name of each variable,

- The purpose of data management is to make a change or changes to this Excel table, for example,
  - You can change the value of a variable for some or all observations
  - You can change the name or attribute of a variable
  - You can add new variables, new observations, or both.

Using SAS to Read in Data

- If you have a SAS system file (i.e., auto.sas7bdat) stored in a directory (c:\temp\in) and you want to save it to another directory (c:\temp\out)

```
LIBNAME in  “c:\temp\in”;
LIBNAME out “c:\temp\out”;
DATA  out.auto2;
SET  “in.auto”;
RUN;
```

- If you have SAS export file (i.e. auto.xpt or “auto.exp) stored in a directory (c:\temp\in) and you want to save it to another directory (c:\temp\out)

```
LIBNAME in xport “c:\temp\in\auto.xpt”;
LIBNAME out “c:\temp\out”;
DATA out.auto2;
SET in.auto;
RUN;
```

Use SAS to Read in Data (Continued)

- If you have raw data file (i.e., auto.dat):

```
LIBNAME    new c:\temp\out;
FILENAME   datafile C:\temp\in\auto.dat;
DATA       new.auto;
INFILE     datafile LRECL=59;
INPUT      make $ 1-17          price  18-22   mpg  23-24    rep78  25      headroom  26-28
           Trunk  29-30         weight  31-34  length  35-37  turn  38-39   displacement  40-42
           gear_ratio  43-58    foreign  59 ;
LABEL      make = "Make and Model“
           price = "Price"
           mpg = "Mileage (mpg)"
           rep78 = "Repair Record 1978“
           headroom = "Headroom (in.)“
           trunk = "Trunk space (cu. ft.)“
           weight = "Weight (lbs.)“
           length = "Length (in.)“
           turn = "Turn Circle (ft.) “
           displacement = "Displacement (cu. in.)“
           gear_ratio = "Gear Ratio“
           foreign = "Car type" ;
RUN;
```

Use SAS to Read in Data (Continued)

```
The relation between the command file and the raw
   data file


12345678901234567890123456789012345678901234567890123456789
AMC Concord       40992232.5112930186401213.57999923706050
```

- if the data are in Excel, Stata, or SPSS format,
  - Use Stat/transfer software to translate the file into a SAS system file.

- If you need to input the data yourself,
  - Type data into an excel file, and use Stat/Transfer to transfer it into a SAS file

Take a Look at the Data

- Find the attribute of data

```
PROC CONTENTS DATA = new.auto position;
RUN;
```

- Take a look at the values of a variable
  - Summary statistics for numeric variable

```
PROC MEANS DATA = new.auto;
VAR price mpg;
RUN;
```

  - Frequencies for both numeric and string variables

```
PROC FREQ DATA = new.auto;
TABLES make price mpg;
RUN;
```

- Take a Look at the variable for some observations

```
PROC PRINT DATA = new.auto (firstobs = 1 obs = 60);
VAR make price mpg foreign;
WHERE (mpg <=20 and foreign =0);
RUN;
```

Take a Look at Data (Continued)

- SAS operators

Table 2. A list of SAS operators			
Comparison	symbol	text	Meanings
	=	eq	Equal
	^=	ne	Not equal
	>	gt	Greater than
	<	lt	Less than
	>=	ge	greater than or equal to
	<=	le	less than or equal to
Logical			
	&	and	both
	|	or	either
	^	not	not true
Arithmeic			
	+		Addition
	-		subtracction
	*		Multiplication
	/		Division
	**		Exponentiation

Take a Look at Data (Continued)

- The order of priorities of SAS operators:
  - All comparison Operators are equal
  - All Logical operators are equal.
  - Within Arithmetic Operators (Exponentiation > Multiplication or Division > Addition or Subtraction.
  - Among these three types of operators, Arithmetic operators > Logical operators > Comparison Operators
  - If you don't know how SAS will decide the order of operators, you can use parenthesis to make sure of it.

Change Orders of Observations or Variables

- The SAS command file to sort the observation

```
PROC SORT DATA=new.auto OUT=new_auto_s;
BY mpg;
RUN;
```

- The SAS commands to check if the data had been sorted

```
PROC PRINT DATA=new.auto;
VAR mpg
RUN;
PROC PRINT DATA=new.auto_s;
VAR mpg;
RUN;
```

Change Orders of Observations or Variables (Cont.)

- The SAS command to change the order of variable:

```
DATA new.auto3;
RETAIN      foreign make price mpg rep78
            headroom trunk weight length turn
            displacement gear_ratio;
SET new.auto;
RUN;
```

- The SAS command for checking if the order of variable has been changed

```
PROC CONTENTS DATA = new.auto3 position;
RUN;
```

Modify Variables

- Change the name of a variable

```
DATA new.auto2;
SET new.auto (rename=(mpg=mpg2 price=price2));
RUN;

DATA new.auto2;
SET new.auto;
RENAME mpg =mpg2 price=price2;
RUN;
```

- Change the value of a variable

```
DATA new.auto2;
SET new.auto;
repair = .;
IF (rep78=1) OR (rep78=2) THEN repair = 1;
IF (rep78=3) THEN repair = 2;
IF (rep78=4) OR (rep78=5) THEN repair = 3;
RUN;
PROC FREQ DATA=new.auto2;
TABLES rep78* repair;
RUN;
```

Modify Variables (Continued)

- Change the numeric variables to string variables and vice versa

```
DATA new.auto2;
SET new.auto;
s_mpg = put(mpg, $8.);
n_mpg = input(s_mpg,8.0);
label s_mpg = "mpg as a string variable"
label n_mpg = "mpg as a numeric variable";
RUN;
```

- You can use the PROC CONTENTS command to check if a variable is a numeric or string variable

```
PROC CONTENTS DATA = new.auto2 position;
RUN;
```

Add Labels

- Three types of labels: data labels, variable labels, and value labels
  - Add labels to the data and variables

```
DATA new.auto2 (LABEL = "new auto data");
SET new.auto;
LABEL rep78 = "Repair Record in 1978"
            mpg = "Miles Per Gallon"
            foreign= "foreign or domestic car";
RUN;
```

  - Check the labels of the data and variables

```
PROC CONTENTS DATA =new.auto2 POSITION;
RUN;
```

Add Labels (Continued)

- Add value labels

```
PROC FORMAT;
VALUE forgnf 0="domestic" 1="foreign" ;
VALUE $makef "AMC" ="American Motors" "Buick" ="Buick (GM)"
   "Cad." ="Cadillac (GM)" "Chev." ="Chevrolet (GM)" "Datsun"
   ="Datsun (Nissan)";
RUN;
```

- Use value labels

```
PROC FREQ DATA=auto2;
FORMAT foreign forgnf. make $makef.;
TABLES foreign make;
RUN;
```

Create New Variables

- Some examples of creating new variables:

```
DATA new.auto2;
SET new.auto;
auto=1;
lag_mpg = lag(mpg);
dif_mpg = dif(mpg);
If  rep78 >=3 then dummy =1;
else if rep78 <3 and rep78 ne . then dummy =0;
else dummy =.;
dummy2 = dummy*2;
interact = foreign*price;
RUN;
```

- Check the accuracy of these new variables

```
PROC PRINT DATA = new.auto2;
VAR mpg lag_mpg dif_mpg rep78 dummy dummy2 foreign price interact;
RUN;
```

Merge Data

- Before you merge data files
  - How many data files do you want to merge them together?
  - Do these data sets have variables with the same name?
  - What is the ID variable used to merge these files?
  - Does the ID variable uniquely identify each observation in each data set? You can do one-to-one or one-to-many merge, but not many-to-many merge.
- Steps of merging data
  - Sort the first data file, based on the ID variable.
  - Sort the second data file, based on the ID variable.
  - Merge two data sets, with the use of the ID variable.

Merge Data (Continued)

- The example of one-to-one merge

Table 3 The first sample data, data1)

make	model	price	mpg	rep78	headroom
AMC	Concord	4099	22	3	2.5
AMC	Pacer	4749	17	3	3
AMC	Spirit	3799	22		3
Buick	Century	4816	20	3	4.5
Buick	Electra	7827	15	4	4
Buick	LeSabre	5788	18	3	4
Buick	Opel	4453	26		3
Buick	Regal	5189	20	3	2
Buick	Riviera	10372	16	3	3.5
Buick	Skylark	4082	19	3	3.5

Table 4. The second sample data (i.e., data2)

make	model	trunk	weight	length	turn	displacement	gear_ratio
Buick	Opel	10	2230	170	34	304	2.87
Buick	Regal	16	3280	200	42	196	2.93
Buick	Riviera	17	3880	207	43	231	2.93
Buick	Skylark	13	3400	200	42	231	3.08
Cad.	Deville	20	4330	221	44	425	2.28
Cad.	Eldorado	16	3900	204	43	350	2.19
Cad.	Seville	13	4290	204	45	350	2.24
Chev.	Chevette	9	2110	163	34	231	2.93
Chev.	Impala	20	3690	212	43	250	2.56
Chev.	Malibu	17	3180	193	31	200	2.73
Chev.	Monte Carlo	16	3220	200	41	200	2.73
Chev.	Monza	7	2750	179	40	151	2.73
Chev.	Nova	13	3430	197	43	250	2.56

Merge Data (Continued)

Expected result of one-to-one merge

Table 5. Merged data file

make	model	price	mpg	rep78	headroom	trunk	weight	length	turn	displacement	gear_ratio
AMC	Concord	4099	22	3	2.5						
AMC	Pacer	4749	17	3	3						
AMC	Spirit	3799	22		3						
Buick	Century	4816	20	3	4.5						
Buick	Electra	7827	15	4	4						
Buick	LeSabre	5788	18	3	4						
Buick	Opel	4453	26		3	10	2230	170	34	304	2.87
Buick	Regal	5189	20	3	2	16	3280	200	42	196	2.93
Buick	Riviera	10372	16	3	3.5	17	3880	207	43	231	2.93
Buick	Skylark	4082	19	3	3.5	13	3400	200	42	231	3.08
Cad.	Deville					20	4330	221	44	425	2.28
Cad.	Eldorado					16	3900	204	43	350	2.19
Cad.	Seville					13	4290	204	45	350	2.24
Chev.	Chevette					9	2110	163	34	231	2.93
Chev.	Impala					20	3690	212	43	250	2.56
Chev.	Malibu					17	3180	193	31	200	2.73
Chev.	Monte Carlo					16	3220	200	41	200	2.73
Chev.	Monza					7	2750	179	40	151	2.73
Chev.	Nova					13	3430	197	43	250	2.56

Merge Data (Continued)

- SAS commands for one-to-one merge.

```
PROC SORT DATA=new.data1;
BY make model;
RUN;

PROC SORT DATA=new.data2;
BY make model;
RUN;

DATA new.merged_data;
MERGE new.data1 (IN=data1) new.data2 (IN=data2);
BY make model;
data1 = data1;
data2 = data2;
RUN;
```

Merge Data (Continued)

- The example of one-to-many merge

Table 6. The Make of the Car

make	foreign
AMC	Domestic
Buick	Domestic
Cad.	Domestic
Chev.	Domestic
Dodge	Domestic
Ford	Domestic
Linc.	Domestic
Merc	Domestic
Olds.	Domestic
Plym.	Domestic
Pont.	Domestic
Audi	Foreign
BMW	Foreign
Fiat	Foreign
Honda	Foreign
Mazda	Foreign
Peugeot	Foreign
Renault	Foreign
Subaru	Foreign
Toyota	Foreign
VW	Foreign
Volvo	Foreign

Merge Data (Continued)

Table 7. The Model of the Car

make	model	price	mpg	rep78	headroom	trunk	weight	length	turn	displacement	gear_ratio
AMC	Concord	4099	22	3	2.5	11	2930	186	40	121	3.58
AMC	Pacer	4749	17	3	3	11	3350	173	40	258	2.53
AMC	Spirit	3799	22		3	12	2640	168	35	121	3.08
Buick	Century	4816	20	3	4.5	16	3250	196	40	196	2.93
Buick	Electra	7827	15	4	4	20	4080	222	43	350	2.41
Buick	LeSabre	5788	18	3	4	21	3670	218	43	231	2.73
.											
.											
.											
VW	Dasher	7140	23	4	2.5	12	2160	172	36	97	3.74
VW	Diesel	5397	41	5	3	15	2040	155	35	90	3.78
VW	Rabbit	4697	25	4	3	15	1930	155	35	89	3.78
VW	Scirocco	6850	25	4	2	16	1990	156	36	97	3.78
Volvo	260	11995	17	5	2.5	14	3170	193	37	163	2.98

Merge Data (Continued)

The Expected result of merging the data for the makes and models of the car

Table 8. The Expected Data of the Make and Model of the Car

make	model	price	mpg	rep78	headroom	trunk	weight	length	turn	isplacemei	gear_ratio	foreign
AMC	Concord	4099	22	3	2.5	11	2930	186	40	121	3.58	Domestic
AMC	Pacer	4749	17	3	3	11	3350	173	40	258	2.53	Domestic
AMC	Spirit	3799	22		3	12	2640	168	35	121	3.08	Domestic
Buick	Century	4816	20	3	4.5	16	3250	196	40	196	2.93	Domestic
Buick	Electra	7827	15	4	4	20	4080	222	43	350	2.41	Domestic
Buick	LeSabre	5788	18	3	4	21	3670	218	43	231	2.73	Domestic
Buick	Opel	4453	26		3	10	2230	170	34	304	2.87	Domestic
Buick	Regal	5189	20	3	2	16	3280	200	42	196	2.93	Domestic
Buick	Riviera	10372	16	3	3.5	17	3880	207	43	231	2.93	Domestic
Buick	Skylark	4082	19	3	3.5	13	3400	200	42	231	3.08	Domestic
Cad.	Deville	11385	14	3	4	20	4330	221	44	425	2.28	Domestic
Cad.	Eldorado	14500	14	2	3.5	16	3900	204	43	350	2.19	Domestic
Cad.	Seville	15906	21	3	3	13	4290	204	45	350	2.24	Domestic
.												
.												
.												
VW	Dasher	7140	23	4	2.5	12	2160	172	36	97	3.74	Foreign
VW	Diesel	5397	41	5	3	15	2040	155	35	90	3.78	Foreign
VW	Rabbit	4697	25	4	3	15	1930	155	35	89	3.78	Foreign
VW	Scirocco	6850	25	4	2	16	1990	156	36	97	3.78	Foreign
Volvo	260	11995	17	5	2.5	14	3170	193	37	163	2.98	Foreign

Merge Data (Continued)

- SAS commands for one-to-many merge.

```
PROC SORT DATA=new.auto_make;
BY make;
RUN;

PROC SORT DATA=new.auto_model;
BY make;
RUN;

DATA new.auto;
MERGE       new.auto_make (IN=auto_make)
            new.auto_model (IN=auto_model);
BY make;
auto_make = auto_make;
auto_model = auto_model;
RUN;
```

Create a Subset of Data

- Keep certain variables

```
DATA new.auto2;
SET new.auto;
KEEP make mpg;
RUN;
```

- Delete certain variables

```
DATA new.auto2;
SET  new.auto;
DROP make mpg;
RUN;
```

Create a Subset of Data (Cont.)

- Keep certain respondents

```
DATA new.auto2;
SET  new.auto;
IF REP78 ^= . ;
RUN;
```

- Delete respondents

```
DATA new.auto2;
SET  new.auto;
IF REP78 = .  THEN DELETE;
RUN;
```

Strengths and Weaknesses of SAS

- Strengths
  - SAS has the ability to handle large data sets
  - Can open multiple data sets at the same time
  - Separate the data step from the analysis step
- Weaknesses
  - It could be difficult to use SAS to change the data format from wide to long or vise versa
  - SAS is not based on open source codes, meaning that SAS does not quickly incorporate new analysis into it. For example, combining survey weights with event history analysis is available in Stata, but not available in SAS yet.
  - It still takes a lot of time to use SAS to create graphs.
  - The documentations of SAS are sometimes hard to understand
  - Rules may change from one version of SAS to another or from one platform to another
  - Needs to move back and forth between the data step and the analysis step

Conclusion

- SAS is a powerful tool for analyzing large data sets
- You can learn SAS by writing a few SAS command files first. For example, you could learn few key commands first and then explore various options of the commands.
- Be aware of all the rules in SAS programming.
- Beware of the logical flow in using SAS
  - You need to read a data before you can do anything about it.
  - You need to sort the data sets first before you can merge them together
  - You need to have data ready before analyzing them
- Several things we have not discussed, but are very important
  - The do loop
  - Array and Macro
  - Other ways of manipulate string variables.
  - Manipulate date variables

Where to Find Help

- Useful websites
  - SAS website (http://support.sas.com/documentation/cdl/en/lrdict/63026/HTML/default/viewer.htm#a000293668.htm)
  - UCLA (http://www.ats.ucla.edu/stat/SAS/)
  - University of Indiana (http://www.indiana.edu/~statmath/stat/sas/win/index.html)
- User group (http://support.sas.com/usergroups/index.html)
- CFDR programming support
  - Hsueh-Sheng Wu @ 372-3119 or wuh@bgsu.edu