STAT 425 and STAT 625
Statistical Software

Lecture 8
Subsetting and Combining SAS Data Sets

Saida Zardi

Subsetting a SAS Data Set

To subset a SAS data set, means to select observations from one data set by defining selection criteria usually using a WHERE or subsetting IF statement.

Subsetting a SAS Data Set

Example:

```
data Females;
    set Learn.Survey;
    where Gender = 'F';
run;

title 'Listing of data set Females';
proc print data=Females noobs;
run;
```

Listing of data set Females

ID	Gender	Age	Salary	Ques1	Ques2	Ques3	Ques4	Ques5
002	Female	51+	$76,123	Agree	Strongly agree	Disagree	Strongly disagree	Strongly disagree
004	Female	51+	$128,000	Strongly agree	No opinion	Disagree	Disagree	Agree
007	Female	30 to 50	$76,100	Strongly agree	No opinion	Agree	No opinion	No opinion

Subsetting a SAS Data Set

The subset data set contains all the variables of the original SAS data set.

To select some variables only, use the options

KEEP = *variable - list* : tells SAS which variables to keep

or

DROP = *variable-list* : tells SAS which variables to drop

KEEP and DROP options can be used in a DATA step or in a PROC step.

Example: Using a Keep= *variable-list*

```
data Females;
   set Learn.Survey(keep=ID Gender Age Ques1-Ques5);
   where Gender = 'F';
run;
```

Using Keep=Data set Option

ID	Gender	Age	Ques1	Ques2	Ques3	Ques4	Ques5
002	Female	51+	Agree	Strongly agree	Disagree	Strongly disagree	Strongly disagree
004	Female	51+	Strongly agree	No opinion	Disagree	Disagree	Agree
007	Female	30 to 50	Strongly agree	No opinion	Agree	No opinion	No opinion

NOTE: the KEEP and DROP options do not change the input data sets, they change only what is read from the input data sets.

There are KEEP and DROP statement that are similar to the KEEP and DROP options, but:

- They can be used only in a DATA step
- They apply to all data sets named in the DATA step
- They are less efficient when dealing with large data sets.

Creating More than one subset data set in one data step

It's possible to create multiple SAS data sets from one input data set.

Example:

```
data Males Females;
    set Learn.Survey;
    if Gender = 'F' then output Females;
    else if Gender = 'M' then output Males;
run;
```

Note: The OUTPUT statement must be followed by the name of the data set to be created, otherwise SAS will output the observations to all the data sets listed in the DATA statement.

The output will provide two data sets one for:

Females

ID	Gender	Age	Salary	Ques1	Ques2	Ques3	Ques4	Ques5
002	Female	51+	$76,123	Agree	Strongly agree	Disagree	Strongly disagree	Strongly disagree
004	Female	51+	$128,000	Strongly agree	No opinion	Disagree	Disagree	Agree
007	Female	30 to 50	$76,100	Strongly agree	No opinion	Agree	No opinion	No opinion

and one for

Males

ID	Gender	Age	Salary	Ques1	Ques2	Ques3	Ques4	Ques5
001	Male	Less than 30	$28,000	Strongly disagree	Disagree	Strongly disagree	Disagree	No opinion
003	Male	30 to 50	$36,500	Disagree	Disagree	Disagree	Disagree	Strongly disagree
005	Male	Less than 30	$23,060	No opinion	No opinion	No opinion	Agree	Disagree
006	Male	51+	$90,000	Disagree	No opinion	Strongly agree	Agree	No opinion

Adding observations

To create a single data set from several similar data sets, list as many data sets you want in a **SET** statement and SAS will add all the observations together and form one data set.

Example: Concatenating data sets

All observations from data set One are followed by the observations in data set Two

Data set One

ID	Name	Weight
7	Adams	210
1	Smith	190
2	Schneider	110
4	Gregory	90

Data set Two

ID	Name	Weight
9	Shea	120
3	O'Brien	180
5	Bessler	207

```
data One_Two;
    set One Two;
run;
```

Data set One_Two

Obs	ID	Name	Weight
1	7	Adams	210
2	1	Smith	190
3	2	Schneider	110
4	4	Gregory	90
5	9	Shea	120
6	3	O'Brien	180
7	5	Bessler	207

What if the data sets don't contain exactly the same variables?

Example:

Data set One

ID	Name	Weight
7	Adams	210
1	Smith	190
2	Schneider	110
4	Gregory	90

Data set Three

Obs	ID	Gender	Name
1	10	M	Horvath
2	15	F	Stevens
3	20	M	Brown

```
data One_Three;
    set One Three;
run;
```

Data set One_Three

Obs	ID	Name	Weight	Gender
1	7	Adams	210	
2	1	Smith	190	
3	2	Schneider	110	
4	4	Gregory	90	
5	10	Horvath	.	M
6	15	Stevens	.	F
7	20	Brown	.	M

Interleaving Data Sets

Using SET statement with a BY statement

Data set One

ID	Name	Weight
7	Adams	210
1	Smith	190
2	Schneider	110
4	Gregory	90

Data set Two

ID	Name	Weight
9	Shea	120
3	O'Brien	180
5	Bessler	207

```
proc sort data=One;
   by ID;
run;
proc sort data=Two;
   by ID;
run;

data Interleave;
   set One Two;
   by ID;
run;
```

Listing of Interleave

Obs	ID	Name	Weight
1	1	Smith	190
2	2	Schneider	110
3	3	O'Brien	180
4	4	Gregory	90
5	5	Bessler	207
6	7	Adams	210
7	9	Shea	120

Combining Detail and Summary Data

```
proc means data=Learn.Blood noprint;
   var Chol;
   output out = Means(keep=Chol_Mean)
          mean = / autoname;
run;
```

Listing of Means

Chol_Mean
201.435

Using the SET statement Conditionally

```
data Percent;
   set Learn.Blood(keep=Subject Chol);
   if _n_ = 1 then set Means;
   PercentChol = Chol / Chol_Mean;
   format PercentChol percent8.;
run;
```

The Five First Observations of Data set Percent

Subject	Chol	Chol_Mean	PercentChol
1	258	201.435	128%
2	.	201.435	.
3	184	201.435	91%
4	.	201.435	.
5	187	201.435	93%

Merging Two Data Sets

Using the MERGE statement to combine two or more SAS data sets.

Listing of Data Set Hours

ID	JobClass	Hours
1	A	39
4	B	44
9	B	57
5	A	35

Listing of Data Set Employee

ID	Name
7	Adams
1	Smith
2	Schneider
4	Gregory
5	Washington

Merging Two Data Sets

- First sort each data set by the variable or variables that link both data sets.
- In this example it's the variable ID
- Second, use the name of the data sets in the MERGE statement which needs to be followed with a BY statement, as in the example:

```
proc sort data=Employee;
    by ID;
run;
proc sort data=Hours;
    by ID;
run;
data Combine;
    merge Employee Hours;
    by ID;
run;
```

Merging Two Data Sets

When the By variable is present in both data sets, the merged observation contains the values from both data sets.
When the BY variable is missing from **Employee** data set, the variable **Name** will have missing values.
When the BY variable is missing from the **Hours** data set, the JobClass and Hours will have missing values.

Listing of Data Set Combine

ID	Name	JobClass	Hours
1	Smith	A	39
2	Schneider		.
4	Gregory	B	44
5	Washington	A	35
7	Adams		.
9		B	57

NOTE: What if we omit the BY statement in the previous program? the result would be completely wrong.

Controlling Observations in a Merged Data Set

SAS Provides a method to control the observations that you want to be in the merged data set.

Using the IN = Option

Example:

```
data New;
   merge Employee(in=In_Employee)
         Hours    (in=In_Hours);
   by ID;
   file print;
   put ID= In_Employee= In_Hours= Name= JobClass= Hours=;
run;
```

IN = data set option follows each dataset name.
variable_name is a temporary variable, equal to 1 if the data set they refer to is making a contribution, and equal to 0, otherwise.
The PUT statement lists the values of these variables.

Controlling Observations in a Merged Data Set

Listing of Data Set Combine

```
ID=1 In_Employee=1 In_Hours=1 Name=Smith JobClass=A Hours=39
ID=2 In_Employee=1 In_Hours=0 Name=Schneider JobClass=  Hours=.
ID=4 In_Employee=1 In_Hours=1 Name=Gregory JobClass=B Hours=44
ID=5 In_Employee=1 In_Hours=1 Name=Washington JobClass=A Hours=35
ID=7 In_Employee=1 In_Hours=0 Name=Adams JobClass=  Hours=.
ID=9 In_Employee=0 In_Hours=1 Name=  JobClass=B Hours=57
```

Employee 1 is in both data sets, so In_Employee=1 and In_Hours is equal to 1, in the first observation.

Employee 2 is in the Employee data set but not the Hours data set so In_Employee =1 and In_Hours=0,…..

Employee 9 has no corresponding ID in the Employee data set and has one in the Hours data set, so In_Employee=0 and In_Hours=1

Controlling Observations in a Merged Data Set

Using the IN=variables to control for the variables written to the output data set → Add an IF statement in the data step

```
data Combine;
   merge Employee(in=In_Employee)
         Hours(in=In_Hours);
   by ID;
   if In_Employee and In_Hours;
run;
```

Listing of Data Set Combine

ID	Name	JobClass	Hours
1	Smith	A	39
4	Gregory	B	44
5	Washington	A	35

More Uses for IN = Variables

Using the IN= variables to check if an ID is missing from a particular data set.

```
data In_Both
   Missing_Name(drop = Name);
   merge Employee(in=In_Employee)
         Hours(in=In_Hours);
    by ID;
    if In_Employee and In_Hours then output In_Both;
    else if In_Hours and not In_Employee then
       output Missing_Name;
run;
```

Listing of Data Set In_Both

ID	Name	JobClass	Hours
1	Smith	A	39
4	Gregory	B	44
5	Washington	A	35

Data set In_Both contains all observations where the ID variable was found in both data sets

Listing of Data Set Missing_Hours

ID	JobClass	Hours
9	B	57

Where Does a Data Step End ?

When any data set reaches an end of file, it signals the end of the DATA step.

Example:

```
data Short;
    input x;
datalines;
1
2
;
data Long;
    input x;
datalines;
3
4
5
6
;
data New;
    set Short;
    output;
    set Long;
    output;
run;
```

In this program, when the end of file on data set Short is encountred, the DATA step ends. So the New data set has only four observations: 1,2,3,4

Merging Two Data Sets with a different By Variable Names

Example:

Listing of Data Set Ernie

EmpNo	Y
123	200
222	205
333	317

Listing of Data Set Bert

ID	X
123	90
222	95
333	100

There's no common name in these two data sets. How can they be merged?

Merging Two Data Sets with a different By Variable Names

Using the RENAME= Data Set Option

Rename = (EmpNo = ID)

So, the data set can be merged on a common variable: ID

```
data Sesame;
   merge Bert
         Ernie(rename=(EmpNo = ID));
   by ID;
run;
```

Listing of Data Set Sesame

ID	X	Y
123	90	200
222	95	205
333	100	317

Merging Two Data Sets with Different By Variable Data Type

Example:

The SS variable is numeric in the data set Division1 and character in the data set Division2

Listing of Data Set Division1

SS	DOB	Gender
111223333	11/14/1956	M
123456789	05/17/1946	F
987654321	04/01/1977	F

Listing of Data Set Division2

SS	JobCode	Salary
111-22-3333	A10	45123
123-45-6789	B5	35400
987-65-4321	A20	87900

Merging Two Data Sets with Different By Variable Data Type

Use of the built-In SAS format: SSN11.

SSN11. Adds 0s and adds dashes as required for SS numbers.

SS becomes a character variable and Division1C and Division2 can be merged.

```
data Division1C;
   set Division1(rename=(SS = NumSS));
   SS = put(NumSS,ssn11.);
   drop NumSS;
run;
```

```
data Both_Divisions;
   ***Note: Both data sets already in order
      of BY variable;
   merge Division1C Division2;
   by SS;
run;
```

Merging Two Data Sets with Different By Variable Data Type

Creating a numeric variable:
using the COMPRESS function then the INPUT function
Then merge both data sets.

```
data Division2N;
    set Division2(rename=(SS = CharSS));
    SS = input(compress(CharSS,,'kd'),9.);
    ***Alternative:
    SS = input(CharSS,comma11.);
    drop CharSS;
run;
```

One to One, One to Many, Many to Many

Example: in the data set Oscar, there are more than one observation for each value of the BY variable ID.

Listing of Data Set Bert

ID	X
123	90
222	95
333	100

Listing of Data Set Oscar

ID	Y
123	200
123	250
222	205
333	317
333	400
333	500

Listing of Data Set Combine

ID	X	Y
123	90	200
123	90	250
222	95	205
333	100	317
333	100	400
333	100	500

One to One, One to Many, Many to Many

Listing of Data Set Many_One

ID	X
123	90
123	80
222	95
333	100
333	150
333	200

Listing of Data Set Many_Two

ID	Y
123	3
123	4
123	5
222	6
333	7
333	8

Listing of Data Set Many_To_Many

ID	X	Y
123	90	3
123	80	4
123	80	5
222	95	6
333	100	7
333	150	8
333	200	8

Updating a Master File from a Transaction file

Merging two data sets to update values from one data set to the other:
Using the UPDATE statement

Listing of Data Set Prices

ItemCode	Description	Price
150	50 foot hose	19.95
175	75 foot hose	29.95
200	greeting card	1.99
204	25 lb. grass seed	18.88
208	40 lb. fertilizer	17.98

Listing of Data Set New15Dec2017

ItemCode	Price
175	25.11
204	17.87
208	.

Updating a Master File from a Transaction file

```
proc sort data=Prices;
    by ItemCode;
run;
```

```
proc sort data=New15Dec2017;
    by ItemCode;
run;
```

```
data Prices_15dec2017;
    update Prices New15Dec2017;
    by ItemCode;
run;
```

Updating a Master File from a Transaction file

Updated data set

Listing of Data Set Prices_15Dec2017

ItemCode	Description	Price
150	50 foot hose	19.95
175	75 foot hose	25.11
200	greeting card	1.99
204	25 lb. grass seed	17.87
208	40 lb. fertilizer	17.98