**Chapter XI - Reading SAS Data Sets**

1. Reading a Single Data Set

Basic code:

**DATA** *SAS-data-set***;**

**SET** *SAS-data-set***;**

<*more SAS statements*>

**RUN;**

* + - * *SAS-data-set* in the DATA statement is the name (libref.filename) of the SAS data set to be created
      * *SAS-data-set* in the SET statement is the name (libref.filename) of the SAS data set to be read.

Eg:

**LIBNAME** sasuser "C:\Users\name\sasuser\";

**LIBNAME** Men50 "C:\Users\name\sasuser\Men50";

**DATA** Men50.males;

**SET** sasuser.admit;

**WHERE** sex='M' **AND** age>50;

**RUN**;

1. Manipulating Data

specify DROP= and KEEP= in either the DATA statement or the SET statement, depending on whether you want to drop variables onto output or input

* If you never reference certain variables and you don't want them to appear in the new data set, use a DROP= option in the SET statement.
* If you do need to reference a variable in the original data set (in a subsetting IF statement, for example), you can specify the variable in the DROP= or KEEP= option in the DATA statement. Otherwise, the statement that references the variable uses a missing value for that variable.

Eg:

**DATA** lab23.drug1h(**DROP**=placebo);

**SET** research.cltrials(**DROP**=triglycerides uricacid);

**IF** placebo='YES';

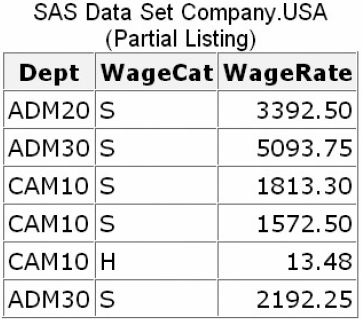
**RUN**;

1. Using **BY**-Group Processing (Chapter 8, page 2)（50题，Q10）

The **DATA** step creates two temporary variables for each **BY** variable. One is named **FIRST**.variable, where variable is the name of the **BY** variable, and the other is named **LAST**.variable. Their values are either 1 or 0. **FIRST**.variable and **LAST**.variable. identify the first and last observation in each **BY** group.

|  |  |
| --- | --- |
| **This variable . . .** | **Equals . . .** |
| FIRST.variable | 1 for the first observation in a BY group |
|  | 0 for any other observation in a BY group |
| LAST.variable | 1 for the last observation in a BY group |
|  | 0 for any other observation in a BY group |

Eg (with single variable):

You want to compute the annual payroll by department. Assume 2,000 work hours per year for hourly employees.

* Before computing the annual payroll, you need to group observations by values of the variable Dept.

Program:

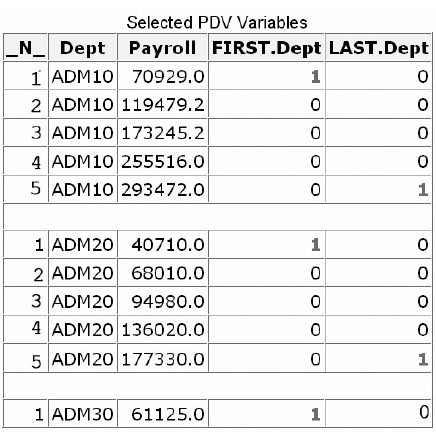
**PROC** **SORT** data=company.usa **OUT**=work.temp;

**BY** dept;

**RUN**;

**DATA** company.budget(**KEEP**=dept payroll);

**SET** work.temp;

**BY** dept;

**IF** wagecat='S' **THEN** Yearly=wagerate\*12;

ELSE IF wagecat='H' **THEN** Yearly=wagerate\*2000;

**IF** FIRST.dept **THEN** Payroll=0;

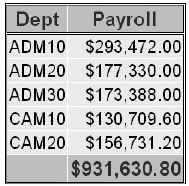
payroll+yearly;

**IF** LAST.dept;

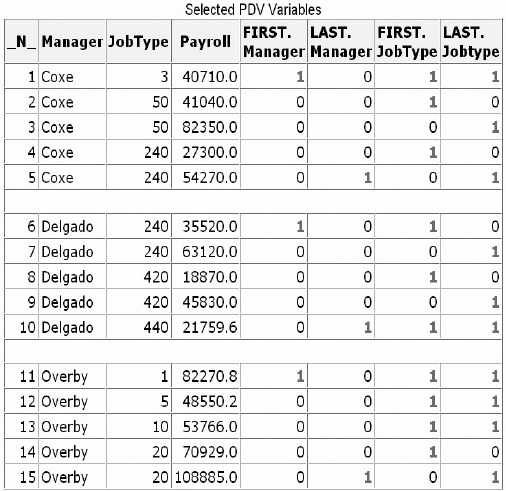
**RUN**;

* The variable name Dept has been appended to FIRST. and LAST, the program data vector (PDV) as the Company.Budget data set is created. See figure on right:
* **PROC** **PRINT** data=company.budget noobs;

**SUM** payroll;

**FORMAT** payroll dollar12.2;

**RUN**;



Eg (with multiple variable):

**PROC** **SORT** data=company.usa **OUT**=work.temp2;

**BY** manager jobtype;

**DATA** company.budget2(**keep**=manager jobtype payroll);

**SET** work.temp2;

**BY** manager jobtype;

**IF** wagecat='S' **THEN** Yearly=wagerate\*12;

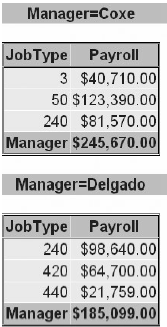
**ELSE** **IF** wagecat='H' **THEN** Yearly=wagerate\*2000;

**IF** first.jobtype **THEN** Payroll=0;

payroll+yearly;

**IF** last.jobtype;

**RUN**;

**PROC PRINT** data=company.budget2 noobs;;

**BY** manager;

**VAR** jobtype;

**SUM** payroll;

**WHERE** manager **IN** ('Coxe','Delgado');

**FORMAT** payroll **DOLLAR**12.2;

**RUN**;

1. Reading Observations Using Direct Access

* **POINT=** option

To access observations directly by their observation number, you use the POINT= option in the SET statement.

* Basic code:

**POINT=variable;**

Variable: specifies a temporary numeric variable that contains the observation number of the observation to read must be given a value before the SET statement is executed.

* **STOP** statement

Preventing Continuous Looping with POINT=

* **OUTPUT** statement
* To override the default way in which the DATA step writes observations to output.
* Placing an explicit OUTPUT statement in a DATA step overrides the automatic output, so that observations are added to a data set only when the explicit OUTPUT statement is executed.
* Basic code:

**OUTPUT** <SAS-data-set(s)>;

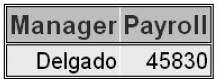
SAS-data-set(s) names the data set(s) to which the observation is written. All data set names that are specified in the OUTPUT statement must also appear in the DATA statement.

* Eg:

Read only the fifth observation from a data set. In the DATA step, the value 5 is assigned to the variable ObsNum. The POINT= option reads the value of ObsNum to determine which observation to read from the data set

**DATA** work.getobs5;

ObsNum=5;

**SET** company.usa(**KEEP**=manager payroll) **POINT**=obsnum; **Result**

**OUTPUT**;

**STOP**;

**RUN**;

**PROC** **PRINT** data=work.getobs5 **NOOBS**;

**RUN**;

* When DATA statement contains two data set names, the DATA step will create both data sets, but only the data set that is named in the OUTPUT statement will contain output

Eg:

**DATA** empty full;

**SET** company.usa;

**OUTPUT** full;

**RUN**;

1. Detecting the End of a Data Set

* **END** statement

To create a temporary numeric variable whose value is used to detect the last observation

Basic code:

**END**=variable

variable creates and names a temporary variable that contains an end-of-file marker. The variable, which is initialized to 0, is set to 1 when the SET statement reads the last observation of the data set.

注意：Do not specify **END**= with **POINT**=. **POINT**= reads only a specific observation, so the last

observation in the data set is not encountered.

 Eg:

**DATA** work.addtoend(**DROP**=timemin timesec);

**SET** sasuser.stress2(**KEEP**=timemin timesec) **END**=last;

TotalMin+timemin;

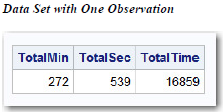
TotalSec+timesec;

TotalTime=totalmin\*60+totalsec;

**IF** **LAST**;

**RUN**;

**PROC** **PRINT** data=work.addtoend **NOOBS**;

**RUN**;

练习

1. If you submit the following program, which variables appear in the new data set？

**DATA** work.cardiac(**DROP**=age group);

**SET** clinic.fitness(**KEEP**=age weight group);

**IF** group=2 **AND** age>40;

**RUN**;

1. none
2. Weight
3. Age, Group
4. Age, Weight, Group
5. Which of the following programs correctly reads the data set Orders and creates the data set FastOrdr?
6. **DATA** catalog.fastordr(**DROP**=ordrtime);

**SET** july.orders(**KEEP**=product units price);

**IF** ordrtime<4;

Total=units\*price;

**RUN**;

1. **DATA** catalog.orders(**DROP**=ordrtime);

**SET** july.fastordr(**KEEP**=product units price);

**IF** ordrtime<4;

Total=units\*price;

**RUN**;

1. **DATA** catalog.fastordr(**DROP**=ordrtime);

**SET** july.orders(**KEEP**=product units price ordrtime);

**IF** ordrtime<4;

Total=units\*price;

**RUN**;

1. none of the above
2. Which of the following statements is *false* about BY-group processing?

When you use the BY statement with the SET statement:

1. The data sets listed in the SET statement must be indexed or sorted by the values of the BY variable(s).
2. The DATA step automatically creates two variables, FIRST. and LAST., for each variable in the BY statement.
3. FIRST. and LAST. identify the first and last observation in each BY group, respectively.
4. FIRST. and LAST. are stored in the data set.
5. There are 500 observations in the data set Usa. What is the result of submitting the following program?

**DATA** work.getobs5;

ObsNum=5;

**SET** company.usa(**KEEP**=manager payroll) **POINT**=obsnum;

**STOP**;

**RUN**;

1. an error
2. an empty data set
3. continuous loop
4. a data set that contains one observation
5. Assuming that the data set Company.USA has five or more observations, what is the result of submitting the following program?

**DATA** work.getobs5;

ObsNum=5;

**SET** company.usa(**KEEP**=manager payroll) **POINT**=obsnum;

**STOP**;

**RUN**;

1. an error
2. an empty data set
3. continuous loop
4. a data set that contains one observation
5. At the start of DATA step processing, during the compilation phase, variables are created in the program data vector (PDV), and observations are set to:
6. blank
7. missing
8. 0
9. there are no observations.
10. The DATA step executes:
11. continuously if you use the POINT= option and the STOP statement.
12. once for each variable in the output data set.
13. once for each observation in the input data set.
14. until it encounters an OUTPUT statement.