NOTE #3: DATA MANIPULATIONS II

3.1 COMBINING DATA SETS

SET data1 data2 …: SET statement is used to stack several data sets; that is combining observations (rows). For example,

```
DATA one;
INPUT name $ score gender $;
DATALINES;
Smith 78 F
Chen 58 F
Rod 69 M
;
DATA two;
INPUT name $ score;
DATALINES;
Park 72
Taylor 81
Lee 57
;
DATA one_two;  SET one two;
RUN;
PROC sort data=one_two; by name; RUN;
PROC PRINT; RUN;
```

You may also want to sort each data first then combine them using

```
SET one two; by name;
```

Conditional SET: In the previous example, suppose that we want to calculate the difference between each score and overall mean. Then, first we can make a new data having the mean score then combine with the original data. Details will be discussed in class. Note that we use PROC MEANS which will be covered more later.

```
/* Example */

PROC means DATA=one_two;
Var score;
Output out=onetwo_out(KEEP=meanscore) Mean = meanscore;
run;
PROC PRINT; RUN;

DATA new;
  SET one_two;
  IF _N_=1 then SET onetwo_out;
  Diff = score - meanscore;
RUN;

PROC PRINT; RUN;
```

One-to One MERGE : Merge statement is used to combine variables (column) from multiple data sets. Let's consider two data sets below. One is the student roster and another is grade data.

```
DATA roster;
INPUT name $ major $ class $;
DATALINES;
Smith CS Senior
Chen STAT Junior
Rod  MATH Sophomore
Park MATH Junior
Taylor SC Sophomore
Lee STAT Senior
;
DATA grade;
INPUT student $ score grade $;
DATALINES;
Smith 65 C
Chen 78 B
Park 95 A
Taylor 87 B
Lee 69 C
;
PROC sort data=roster; by name; run;
PROC sort data=grade; by student; run;

DATA list;
  MERGE roster grade (rename=(student=name); by name;
RUN;

PROC PRINT; RUN;
```

It is important to note that before merging two data sets by name variable data must be sorted by the variable. In case two data sets use difference names of the variable you want to use to merge them, you can use RENAME= option in MERGE statement.

Note also that student name Rod is in the roster data but not in grade data and you may want to keep those records which are in both data sets. You can use IN= option to do this. For example,

```
DATA list;
  MERGE roster(IN=Inroster) grade(IN=Ingrade); by name;
  IF Inroster=1 and Ingrade=1;
RUN;

PROC PRINT; RUN;
```

Here both `Inroster` and `Ingrade` variables (you name them) are logical variables which return 1 if an observation is in the data. For example, `Inroster=1` and `Ingrade=1` means the observation is in both *roster* and *grade* data. `Inroster=1` and `Ingrade=0` means that the observation is in *roster* data but not in *grade* data (the case for the student named Ron). The data *list* in the example above will involved roster and grade information for five students excluding Ron.

One-to-n Merge: Shorter record will repeat. See below.

```
DATA one;
INPUT ID X;
DATALINES;
1 32
3 54
```

```
4 76
5 34
;
DATA two;
INPUT ID A $;
DATALINES;
1 CD
1 DVD
2 USB
3 DVD
3 HD
3 CDR
5 CD
5 DVD
;
DATA combine; MERGE one two; BY ID; RUN;
PROC PRINT; RUN;
```

Note that reversing the order in MERGE statement will not affect the result. Result will be discussed in class.

n-to-n Merge: No problem. They will match line by line assuming data are sorted by the variable used in BY statement.

n-to-m Merge: You may not want to do this. See below.

```
DATA one;
INPUT ID X;
DATALINES;
1 32
1 40
1 99
3 54
4 76
5 50
5 34
;
DATA two;
INPUT ID A $;
DATALINES;
1 CD
1 DVD
2 USB
3 DVD
3 HD
3 CDR
5 CD
5 DVD
;
DATA combine; merge one two; by ID; run;
PROC print; run;
```

```
/* Example 3_1 */


 data stock;
    input Model $ unit Price;
    format Price dollar8.2;
 datalines;
 CRX050 1254 69.50
 KTX012 965 99.99
 DVR010 365 169.85
 SAM055 62  129.95
 LGC052 124 144.59
 PHS199 785 81.99
 ;
 proc sort; by model; run;
 proc print; run;

 DATA purc;
 INFILE DATALINES MISSOVER;
 INPUT (CustID Model) ($) Quantity @; OUTPUT;
 INPUT Model $ Quantity @; OUTPUT;
 INPUT Model $ Quantity ; OUTPUT;
 datalines;
 1001 CRX050 150 SAM055 25
 1002 CRX050 200 LGC052 40 PHS199 30
 1003 SAM055 50
 1004 KTX012 200 LGC052 90 CRX050 30
 1005 CRX050 100
 ;
 DATA purc; set purc;
 if missing(Model) then delete; run;
 proc sort; by Model; run;
 proc print; run;

 Data byModel (drop=Quantity CustID) ; set purc; by model;

 If first.model then unit_out=0;
 unit_out+Quantity;
 if last.model;  run;

 Data comb;
 merge stock byModel; by model;
 if missing(unit_out) then unit_out=0;
 Unit_left=unit-unit_out;
 if Unit_left <=0 then file print;
   put "Please note that the model " model " is out of order";
 output;
 run;

 proc print; run;
```

/* IN-CLASS */

/* Reconsider the sales data in Example 2_2

(a) Create separate data set for each store

(b) For each data calculate the overall average sales a day, identify the most sold item per day and its proportion to overall daily sales

(c) Combine the data in (b) and also identify the most sold item all three stores combined. Your data should look something like

STORE	Daily_sales	Most_Sold/proportion
KENWOOD	$xxxxxxxx	XXXXX / .xxxx
WESTSIDE	$xxxxxxxx	XXXXX / .xxxx
SOUTHHILL	$xxxxxxxx	XXXXX / .xxxx
TOTAL	$xxxxxxxx	XXXXX / .xxxx