

NOTE #5: DISPLAYING DATA**PROC PRINT;**

```

/* Example 5-1 */

PROC FORMAT;
  VALUE $Class 'F'= 'Freshman' 'S'= 'Sophomore'
  'J'= 'Junior' 'Sr'='Seniour';
RUN;

DATA roster;
  INFILE datalines FIRSTOBS=3 MISSOVER;
  INPUT NAME $2-14 Class $15-16 (Score1 Score2 Score3)(3.);

  ARRAY Test (3) Score1-Score3;
  DO i = 1 to 3;
    If missing(Test(i)) then Test(i) =0;
  END;
  DROP i;
  Avg=round(mean(of Score1-Score3),4.1);

Datalines;
STAT101 Elementary Statistic
Fall 2010 Grade Roster
C Gonzalez J 82 60 78
T Chung Sr 92 58 96
I Irwin F 52 69 77
C Reid S 98 92 89
P Lee S 60 48 70
B Washington J 47 63
K Piazza F 87 72 90
M Crafton F 100 74100
A Shulz S 47 61
S Olson F 74 77 75
C Chen S 90 73 68
H Wood F 100 98
;
Title1 " CLASS ROSTER STAT101";
Title2 "===== ";
Title4 "Prof. S KIM";

footnote "footnote appears at the bottom of output page";

PROC sort DATA=roster; by Class; run;

Options nodate pageno=1 linesize=64 pagesize=60;

PROC PRINT DATA=roster LABEL heading=horizontal N="No of Students = ";
  by Class;
  Where Avg GE 70 & Class in ('S' 'F');
  id ;
  var NAME Class Avg;
  sum Avg;
  format Class $Class.;
  label NAME = "Student Name"
  Avg = "Average Score";
run;

```

```

/* Example 5-2*/
/* Consider the store data in Example 2-2 */
/* Assuming the data is saved as SASPRAC.sales */

Proc sort DATA= SASPRAC.sales; By Store; RUN;

Title;
Footnote;
Option nodate pageno=1;

PROC PRINT LABEL DATA= SASPRAC.sales (firstobs=1 obs=3) DOUBLE N="Total
Number of items: ";
id Store;
by Store;
RUN;

```

PROC REPORT: More controls over Proc Print. Usages of variables are DISPLAY (default for character), ANALYSIS (default for numeric), ORDER, GROUP, ACROSS, or COMPUTE. Each usage is illustrated in examples below.

```

/* Example 5-3 */
Option pageno=1 nodate ;

PROC FORMAT;
  VALUE $Class 'F'= 'Freshman' 'S'= 'Sophomore'
  'J'= 'Junior' 'Sr'='Seniour';
RUN;

DATA roster;
  INFILE datalines FIRSTOBS=3 MISSOVER;
  INPUT NAME $2-14 Class $15-16 (Score1 Score2 Score3)(3.);
  LastName=scan(Name,2,' ');
  ARRAY Test (3) Score1-Score3;
  DO i = 1 to 3;
    If missing(Test(i)) then Test(i) =0;
  END;
  DROP i;
  Avg=round(mean(of Score1-Score3),4.1);

Datalines;
STAT101 Elementary Statistic
Fall 2010 Grade Roster
C Gonzalez J 82 60 78
T Chung Sr 92 58 96
I Irwin F 52 69 77
C Reid S 98 92 89
P Lee S 60 48 70
B Washington J 47 63
K Piazza F 87 72 90
M Crafton F 100 74100
A Shulz S 47 61
S Olson F 74 77 75
C Chen S 90 73 68
H Wood F 100 98
;
proc print; run;
Title; footnote;
Title3 " CLASS ROSTER STAT101";
Title4 "===== ";

```

```

Title5 "Prof. S KIM";

PROC REPORT DATA=roster nowd headline ls=84 ps=80 ;
  column Class Name LastName Avg Grade;
  define Class      / order width=8 Format = $Class.;
  define Name       / width=15 spacing = 10;
  define LastName   / order noprint;
  define Avg        / display "Average/Score" width=7;
                        *need display to be used in COMPUTE;
  define Grade      / computed center width=7;

COMPUTE Grade / character length=7;
  if Avg gt 90 then Grade = 'A';
  Else if Avg gt 70 then Grade = 'B';
  Else if Avg gt 50 then Grade = 'C';
  Else if not missing(Avg) then Grade = 'D';
ENDCOMP;

run;

PROC REPORT DATA=roster nowd headline ls=84 ps=80 ;
  column Class Avg ;
  define Class / group Format = $Class.;
  define Avg   / analysis mean "Class Average"; *sum is default;
RUN;

```

NOTE:

1. COLUMN statement is used as VAR statement in PROC PRINT.
2. To subset the data in a report, WHERE statement can be used with PROC REPORT the same way we used in PROC PRINT (see example 1).
3. Both ORDER and GROUP order rows and suppress printing repetitions of the variables. The differences are that GROUP collapses observation with the same values in the variable and gives you summary report, while ORDER provides the list of all observations.
4. ANALYSIS is used to calculate a statistic. The default statistic is SUM. Above example used mean for the summary statistic. Other available statistics are: CSS (correlated sum of squares), USS (uncorrelated SS), CV (coefficient of variation), MAX, MEAN, MIN, N, NMISS (number of observation with missing values), and RANGE.
5. Computed variables are that you define in PROC REPORT and are not in the data set. You have to include the names of the computed variable in the COLUMN statement.
6. You have to use DISPLAY for a numerical variable to be used in COMPUTE. ACROSS functions similarly to GROUP, but it creates the cross variable horizontally while GROUP put the variable vertically. The following example illustrates the use of ACROSS variables.

```

/* Example 5_4 */
/* This example is taken from the book "Learning SAS by Example" */
/* by Ron Cody */

data blood;
  infile 'J:\SASCLASS\blood.txt' trunccover;
  length Gender $ 6 BloodType $ 2 AgeGroup $ 5;
  input Subject
         Gender
         BloodType
         AgeGroup
         WBC

```

```
        RBC
        Chol;
label Gender = "Gender"
      BloodType = "Blood Type"
      AgeGroup = "Age Group"
      Chol = "Cholesterol";
run;

title "Average Blood Counts by Age Group";
proc report data=blood nowd headline;
  column Gender BloodType AgeGroup,WBC AgeGroup,RBC;
  define Gender      / group width=8;
  define BloodType  / group width=8 "Blood Group";
  define AgeGroup   / across "- Age Group -";
  define WBC        / analysis mean format=comma8.;
  define RBC        / analysis mean format=8.2;
run;
```

```
/* In-Class practice (from Cody's book)
```

1. Using the blood data, produce a report showing the mean cholesterol for each combination of Gender and blood type.
2. Using the same data, produce a table of average WBC and RBC counts for each combination of blood type and gender. */