An Introduction to SAS-Lecture 5

Konstantinos Fokianos
University of Cyprus

PROC MEANS

The MEANS and UNIVARIATE procedures are used to produce reports of descriptive summary statistics such as means, standard deviation etc. They can also be used to process the data and produce new data sets containing summary statistics. We will see some different ways in which these procedures can be put to work.

The SAS system also contains a procedure called PROC SUMMARY which is identical to PROC MEANS (with a NOPRINT option). PROC MEANS can do as much as PROC UNIVARIATE but there are some differences, as we shall see next.

Computing Totals

Suppose that you have SAS data set called SALES which contain sales figures for an internet order company. Each record in the data set represents the sales of a single item.The variable ar PO_NUM (purchase order number), ITEM (item description), REGION (region of the country where the item was sold), PRICE and QUANTITY. A partial listing is as follows:

```
PO_NUM ITEM    REGION  PRICE QUANTITY
-------------------------------------
1456   Hammer  NORTH   10    5
1458   Saw     NORTH   15    4
.............................
.............................
1901   Saw     WEST    15    5
```

Computing Totals

You want to see the summary statistics of the QUANTITY sold classified by REGION" and ITEM. One way to do this is to use the PROC SORT statement to sort the data by region and item and then use a BY statement in the PROC MEANS. A more efficient approach is to use a CLASS statement to specify the variable that you want to sort in the PROC MEANS.

We also want to create an output data set which contains the totals (SUM) of the quantities for the various region-items categories.

Computing Totals

```
PROC MEANS DATA=SALES;
   TITLE 'Sample Output from PROC MEANS';
   CLASS  REGION ITEM;
   VAR    QUANTITY;
   OUTPUT OUT=QUAN_SUM SUM=TOTAL;
RUN;

PROC PRINT DATA=QUAN_SUM;
   TITLE 'Summary Data Set';
RUN;
```

Computing Totals

- The procedure automatically computes the following: N, Mean, Standard Deviation, Minimum and Maximum.
- The `CLASS` statement breaks down the data by the variables specified. Then `PROC MEANS` also calculates the number of observations per group (`N Obs`). `N` is the number of non-missing data.
- The `VAR` statement specifies the variables that we want to study.

Computing Totals

- The `OUT` option in the `OUTPUT` statement specifies the names of the output data set you want to create. Numerous statistics (we asked for `SUM`) can be included in this data set. Typical statistics are `N=`, `MEAN=`, and `SUM=`. Following each of these keywords is a list of variables to be included in the newly created data set for each of the variables specified in the `VAR` statement.
- In order to see the results, you need to add a `PROC PRINT` procedure. `PROC MEANS` produces the summary results (to be included in the output data set) plus the variables `_TYPE_` and `_FREQ_`.

Computing Totals

- The first observation has `_TYPE_=0` and represents the entire population. The value `TOTAL` is equal to 143 and this is the sum of `QUANTITY` for all regions and items. The `_FREQ_` value shows that there 11 observations contributing to this sum.
- The next three lines (`_TYPE_=1`) gives the sums for each level of the last (rightmost) `CLASS` variable, `ITEM`, across regions. Here `_FREQ_` shows how many orders were placed for each item and `TOTAL` gives the value of how many `ITEM` were sold all together.
- The next four lines show the same information but for the variable `REGION`. The remaining lines are the totals for each combination of all the `CLASS` variables.

Computing More than One Statistic

If you have more than one variable in your VAR list, the variables names in the output statement following following N=, MEAN=, SUM= etc represent the variables in the output data set that contain the values of that statistics for each variable in the VAR list in the order which they are listed.
Here is a short example:

```
PROC MEANS DATA=ORIGDATA;
   CLASS A B;
   VAR   X Y Z;
   OUTPUT OUT  = STATS
          N    = NUM_X   NUM_Y   NUM_Z
          MEAN = MEAN_X  MEAN_Y  MEAN_Z
          SUM  = TOT_X   TOT_Y   TOT_Z;
RUN;
```

Computing More than One Statistic

The variables NUM_X, NUM_Y and NUM_Z in the output data set STATS contain the values for the number of non missing observations for the variables X, Y and Z.
Similarly, the other variables contain the means and the sums of the variables X,Y and Z.

Computing Unweighted Summary Statistics

Suppose that you have data which contains blood pressure readings for a number of subjects but there are a variable number of observations per subject per year. You want to calculate a mean value for your measurements per year. Variables are the SUBJ, YEAR, SBP (systolic blood pressure) and DBP (diastolic blood pressure).

```
SUBJ   YEAR   SBP   DBP
------------------------
1   1950    130   80
1   1950    132   82
1   1951    140   86
2   1950    118   72
2   1950    120   74
2   1952    122   76
2   1952    116   74
```

Computing Unweighted Summary Statistics

You first have to compute the mean SBP and DBP for each subject for each year and put these values to a new data set. You use this new data set to compute yearly means over all subjects. In the following program, we calculated unweighted yearly means.

```
PROC MEANS DATA=PRESSURE NOPRINT NWAY;

   CLASS  SUBJ YEAR;
   VAR    SBP DBP;
   OUTPUT OUT=MEANOUT MEAN=;
RUN;

PROC PRINT DATA=MEANOUT;

RUN;
```

Computing Unweighted Summary Statistics

- The NOPRINT option tells system not to print the results. However we use a PROC PRINT later; this is simply for illustration.
- In this example you do no need to specify the variables after the MEAN in the OUTPUT statement. This results in the mean SBP and DBP in the output data set having the same variable names as those indicated in the VAR statement. TRY TO AVOID THIS because it is rather confusing and probably you will need more than one statistic to compute.
- The output (when compared to previous ones) is different because of the NWAY option. You use this option to tell SAS to produce output for only the highest level of class intercations.

Computing Unweighted Summary Statistics

Now we are ready to complete our task by manipulating the new data set that we created in the previous steps.

```
PROC MEANS DATA=MEANOUT MEAN MAXDEC=2;
   TITLE 'Averages Computed from Person Yearly Means';
   CLASS YEAR;
   VAR SBP DBP;
RUN;
```

Computing Unweighted Summary Statistics

The resulting output shows the mean of SBP and DBP for each year, computed from the yearly means of each subject. The MAXDEC=2 limits the output values to two decimal places.

YEAR	N	Obs	Variable	Mean
1950	2		SBP	125.00
			DBP	77.00
1951	1		SBP	140.00
			DBP	86.00
1952	1		SBP	119.00
			DBP	75.00

Formatting Output

The data in this example were collected as a part of fund drive. Each observation represents a letter mailed out to a resident asking for a contribution. The variables contained in this data set is NAME, TOWNSHIP and AMOUNT. When the value of AMOUNT is missing, this implies that there was no donation.The goal is to produce a results showing the following information for each township: the number of letters mailed, the total number of donations received, the total amounts received, the average amount received by donation and the average amount received by letter mailed.

Computing Unweighted Summary Statistics

- The statistics N, NMISS and SUM provide you with some of the quantities that you are looking for. The N is output as RETURNED and it will be labeled NUMBER OF OBSERVATIONS. The SUM is output as TOTAL and then is labeled as TOTAL DONATION. NMISS is output as NOT_RETURN.
- In the DATA step, we calculate the rest of the variables we need. You add N (RETURNED) and NMISS (NOT_RETN) to get MAILED (total number of units mailed per township) and label it LETTERS MAILED.
- SUM (TOTAL) is divided by RETURNED (number of donations) to get PER_RETN (MEAN DONATION) and by MAILED (number of units mailed) to get PER_MAIL (MEAN DONATION PER LETTER MAILED).
- The LABEL option of PROC PRINT tells the system to use the right labels.

Computing Percentages

In this example, you want to express individual data values as a percentage of the mean of all subjects in a group. You have a SAS data set containing variables HR (heart rate), SBP (systolic blood pressure) and DBP (diastolic blood pressure) for each subject and you want to express these values as percent of the mean of all subjects.
For example, if the values for heart rate for the group are 80,70 and 60, the group mean is equal to 70 and the first subject's HR value of 80 would yield a percentage score of 114.286 % (100 $\times$ 80/70).

Computing Percentages

```
DATA TEST;
   INPUT HR SBP DBP;
DATALINES;
80 160 100
70 150 90
60 140 80
;

PROC MEANS NOPRINT DATA=TEST;
   VAR HR SBP DBP;
   OUTPUT OUT  = MOUT
          MEAN = MHR MSBP MDBP;
RUN;

DATA PERCENT ;
   SET TEST;
   DROP MHR MSBP MDBP _TYPE_ _FREQ_;
   IF _N_ = 1 THEN SET MOUT;
   HRPER =100*HR/MHR;
   SBPPER=100*SBP/MSBP;
   DBPPER=100*DBP/MDBP;
RUN;

PROC PRINT NOOBS DATA=PERCENT;
   TITLE 'Listing of Percent Data Set';
RUN;
```

Computing Percentages

- You use the NOPRINT option with PROC MEANS because you do not want the the procedure to print anything, only to create a data set of means. In this case, since there is no CLASS variables, the output data set MOUT consists of only one observation.

OBS	_TYPE_	_FREQ_	MHR	MSBP	MDBP
1	0	3	70	150	90

- Now, we want to combine the data from MOUT with the data from every observation in the original data set so that you can divide each single value by the appropriate mean (HR by MHR etc) and then multiply by 100. You create a new data set PERCENT by reading the variables from existing data sets and then creating the right percentages.

Computing Medians

```
DATA TEST;
   INPUT HR SBP DBP;
DATALINES;
80 160 100
70 150 90
60 140 80
;

PROC UNIVARIATE  DATA=TEST;
   VAR HR SBP DBP;
   OUTPUT OUT  = MOUT
          MEAN = MHR MSBP MDBP
  MEDIAN= MEDMHR MEDSBP MEDDBP;
RUN;


PROC PRINT  DATA=MOUT;
   TITLE 'Results from the Univariate Procedure';
RUN;
```

Computing Medians

- The PROC UNIVARIATE enables calculation of medians.
- In addition, it gives various statistical summaries, as we see from the output.