LABEL statement

The LABEL statement assigns SAS variable labels to variables in the output data sets. You can give labels for any number of variables in a single

```
LABEL variable = "label" ;
```

```
data temp;
input x y type $;
label x='Distance Travelled'
           type='Species';
```

PROC FORMAT

There are some build-in SAS format e.g. MMDDYY8. Dollar10. ….

When you have a lot of coded data, you may need to create your own ***custom formats***. For example the age categories teen, adult, and senior can be coded as1, 2, 3. Or you have gender coded as 1-Male, and 2-Female.

PROC FORMAT

The FORMAT procedure can create formats the will later be associated with variables in a FORMAT statement.

```
PROC FORMAT;
            VALUE       SX          1 = “Male”
                                    2 = “Female”
                                                ;
            VALUE       $col        ‘W’ = ‘Moon white’
                                    ‘B’  = ‘Sky Blue’
                                    ‘Y’ = “Sunburst’s Yellow”
                                    ‘G’ = ‘Gray’;
Run;
```

After created the FORMATS, you will need FORMAT statement in the procedure to see the format output. Notice that format names do not end with periods in the value statement, but they do in the FORAMAT statement.

FORMATS

After created the FORMATS, you will need FORMAT statement in the procedure to see the format output. Notice that format names do not end with periods in the value statement, but they do in the FORAMAT statement.

What is the different between FORMAT procedure and FORMAT OPTION?

PROC CONTENTS

Listing the Contents of a SAS Data Set Using PROC CONTENTS

```
PROC CONTENTS          DATA =  data-set name OPTIONS;
Run;
```

OPTIONS:

OUT=*SAS-data-set:* names an output SAS data set.

VARNUM: prints a list of the variable names in the order of their logical position in the data set. By default, the CONTENTS statement lists the variables alphabetically. The physical position of the variable in the data set is engine-dependent.

SHORT: prints only the list of variable names, the index information, and the sort information for the SAS data set

Sorting Your Data with PROC SORT

There are many reasons for sorting your data: to organize data, to combine data sets, or before using BY statement in PROC or DATA statement.

```
PROC SORT   DATA=dataset OPTIONS;
BY var1; or
BY DESCENDING var1;
RUN;
```

You can specify as many variables as you wish. By default SAS will sort the data in ascending order of BY variables. From lowest to highest, from A to Z. We can sort the data from highest to lowest, or from Z to A by adding the key word DESCENDING to the by statement.

Options:

OUT= *SAS-data-set*	Specifies the output data set
NODUPKEY	Delete observations with duplicate BY values

PROC PRINT

PROC PRINT Options;

Statements (Or Options):

Run;

The PRINT procedure is perhaps the most widely used SAS procedure. By default, if you don’t specify the data set, SAS will print the most recently created data set. The following options sometime come in handy:

Options: Data=name (OBS=10): print only first 10 Obs

Statements : ID variable - list; no obs. but ID variable

SUM variable - list; print the sums of the variables

VAR variable – list; specify which variables to print

Summarizing Your Data Using PROC MEANS

The MEANS procedure starts with the keywords PROC MEANS, followed by options listing the statistics you want printed. Following are most commonly used options.

proc means Options;

Optional Statements ;

RUN;

Options:

Data = dataname	
MAX	the maximum value
MIN	the minimum value
MEAN	the mean
MEDIAN	the median
N	number of non-missing values
NMISS	number of missing values
RANGE	the range
STDDEV	the standard deviation
SUM	the sum

Optional Statements:

BY variable - list

CLASS variables - list

VAR variable – list

Output out = data-set

FREQ count; (Only if your data have counting)

PROC UNIVARIATE

PROC UNIVARIATE produces statistics describing the distribution of a single variable. These statistics include the mean, median, mode, standard deviation, skewness, and kurtosis. The common options are PLOT and NORMAL. The NORMAL option produces tests of normality while the PLOT option produces three plots of your data.

PROC BOXPLOT

The BOXPLOT procedure creates side-by-side box-and-whisker plots of measurements organized in groups. A box-and-whisker plot displays the mean, quartiles, and minimum and maximum observations for a group.

The box plot displayed in Example represents summary statistics for the analysis variable kwatts; each of the 10 box-and-whisker plots describes the variable kwatts for a particular day. The plot elements and the statistics they represent are as follows:

The length of the box represents the interquartile range (the distance between the 25th and the 75th percentiles); The dot in the box interior represents the mean; The horizontal line in the box interior represents the median; The vertical lines issuing from the box extend to the minimum and maximum values of the analysis variable

PROC PLOT

The PLOT procedure plots the values of two variables for each observation in an input SAS data set.

```
PROC PLOT DATA = datasetname <option(s)>;
 PLOT plot-request(s) </ option(s)>;
BY <DESCENDING> variable-1;
RUN;
```

To produce a plot of y vs x, in the plot statement use: PLOT y*x;

To change the symbol used for each point in the graph, in the plot statement use

```
                        PLOT y*x = plotsymbol;
```

The plotting symbol can be a variable name not enclosed in single quotes, or an actual keyboard character enclosed in single quotes;

There are some examples of SAS graph:

http://support.sas.com/sassamples/graphgallery/PROC_GPLOT.html

PROC TRANSPOSE

The TRANSPOSE procedure transposes SAS data set, turning observations into variables or variables into observations.

PROC TRANSPOSE DATA = old-data-name OUT = new-data-name;

BY *variable – list*;

ID *variable;*

VAR *variable – list;*

Run;

BY statement: if you have any grouping variables that you want to keep as variables and these variables are included in the transposed data set. Data must sorted by these variables before transpose.

ID statement: The ID statement names the variable whose formatted values will become the new variables names.

VAR statement The VAR statement names the variables whose values you want to transpose.