**Basic SAS Skills**

**Handout #2 – SAS Procedures**

**Summer 2013**

**by Stephen J. Lusch** **| Updated by Pablo C. Machado**

The purpose of this handout is to introduce you to the various Proc statements available to you in the SAS programming language.

**Proc Contents**

Proc Contents can be used to call descriptive information about a dataset. It can be used on datasets either on your computer or on the WRDS server.

Example:

rsubmit;

**proc** **contents** data = comp.funda; **run**; **quit**;

endrsubmit;

For the dataset you will be provided with the name, number of observations, number of variables, and the date created. In addition for each variable you will be given its type (numeric or character), its length, its format for printing, its informat for input, and its label.

**Proc Corr**

Proc Corr is used to calculate the correlation between two variables in your dataset.

Example 1:

**proc** **corr** data = sample; **run**; **quit**;

This basic Proc Corr command will calculate Pearson correlation coefficients for all variables in your dataset.

Example 2:

**proc** **corr** data = sample;

var at txt ni; **run**; **quit**;

By adding the var statement to the procedure you have now limited the variable in which correlation coefficients will be calculated for.

Example 3:

**proc** **corr** spearman data = sample; **run**; **quit**;

By adding spearman to the procedure you are now directing SAS to calculate Spearman correlation coefficients instead of Pearson correlation coefficients.

**Proc Freq**

Proc freq is a procedure used to produce frequency tables of variables in your dataset.

Example 1:

**Proc** **freq** data=sample;

tables year;

**run**; **quit**;

This example will compute a frequency table giving the number of observations for each year.

Example 2:

**Proc** **freq** data=sample;

tables year\*sich;

**run**; **quit**;

The proc freq command also allows you to create frequency tables by group. The procedure in this example will create a frequency table giving the number of observations in each year by SIC code.

**Proc Means**

Proc means is one of the procedures available to calculate descriptive statistics for variables in your dataset.

Example 1:

**Proc** **means** data=sample;

**run**; **quit**;

The basic command will return the mean, the number of non-missing values, the standard deviation, the minimum value, and the maximum value for every numeric variable in the dataset.

Example 2:

**Proc** **means** data=sample MEAN MEDIAN P10 P90 RANGE;

var at ni;

**run**; **quit**;

Example 2 introduces two variations to the proc means procedure.

1. The user can specify which descriptive statistics they would like to receive in their output. In this example the mean, median, 10% percentile, 90% percentile, and range have been asked for.
2. The var statement allows the user to specify which variables they would like the statistics to be calculated for, in contrast to example 1 in which the descriptive statistics were calculated for all numeric variables.

Below is the list of statistics that can be calculated with the proc means procedure.

|  |  |
| --- | --- |
| **Descriptive Statistic Keywords** | |
| CLM – two-sided confidence limits | NMISS – number of missing values |
| CSS – corrected sum of squares | RANGE – the range |
| CV – coefficient of variation | SKEWNESS – skewness |
| KURTOSIS – kurtosis | STDDEV – standard deviation |
| LCLM – lower confidence limit | STDERR – standard error of the mean |
| MAX – maximum value | SUM – the sum |
| Mean – mean | SUMWGT – sum of weight variables |
| MIN – minimum value | UCLM – upper confidence limit |
| MODE – mode value | USS – uncorrected sum of squares |
| N – number of non-missing values | VAR – variance |
| **Quantile Statistic Keywords** | |
| MEDIAN - median | P75 - 75th percentile |
| P1 - 1st percentile | P90 - 90th percentile |
| P5 - 5th percentile | P95 - 95th percentile |
| P10 - 10th percentile | P99 - 99th percentile |
| P25 - 25th percentile | QRANGE – Interquartile range (Q3 – Q1) |
| **Hypothesis Testing Keywords** | |
| PROBT - probability of Student's t | T - Student's t |

**Proc Print**

This procedure prints the contents of a SAS dataset into the output window. When you are working on the WRDS server, this procedure will save you time by printing a limited number of observations from the WRDS dataset so you can verify that you have the data that you need.

Example 1:

**proc** **print** data=sample (obs = **50**);

**run**; **quit**;

This example will print all variables for the first 50 observations of the dataset sample. In addition, you can add the var statement to limit which variables are printed or the by statement if you wish to print data by group, for example by SIC code.

**Proc Reg**

Proc reg is the most basic of the regression procedures available in SAS. This procedure fits linear regression models by least-squares.

Example 1:

**proc** **reg** data=sample;

model *dependent variable = independent variables*;

**run**; **quit**;

In the place of *dependent variable* the user needs to input the variable name for their dependent variable, and in the place of *independent variables* the user lists the variable names of the independent variables separated by a space. For example, a researcher may want to investigate the impact of a number of independent variables (say age, years of education, and gender) on salary. Therefore, the code would be written as follows.

**proc** **reg** data=sample;

model salary = age yrofeduc gender;

**run**; **quit**;

Note that the intercept is automatically included in the proc reg procedure. There are numerous options and other statements that can be added to the proc reg procedure; however for an introduction this is sufficient. Further information can easily be found in various SAS guides and in online resources.

**Proc Sort**

This procedure sorts data according to the variables provided by the user.

Example 1:

**proc** **sort** data=sample;

by gvkey year;

**run**; **quit**;

This example will sort the observations in the dataset sample first by gvkey and then by the year of the observation.

Example 2:

**proc** **sort** data=sample;

by year descending ni;

**run**; **quit**;

By default SAS sorts in ascending order, so if you need to sort a variable in descending order the word "descending" needs to be added before the variable that you would like sorted in descending order. Example 2 will sort the observations in the dataset sample first by total assets in ascending order and then by net income in descending order.

Example 3:

**proc** **sort** nodupkey data=sample;

by gvkey year;

**run**; **quit**;

The nodupkey option for the proc sort procedure is used to eliminate duplicate observations in your sample. This example will sort all observations first by gvkey and then by year, and keep the first one in the dataset. It is somewhat common for users to code in a way that pulls duplicate observations, so this is a good way to purge those observations from your dataset.

**Proc Rank**

Proc rank computes the ranks of the values of numeric variables.

Example 1:

**proc** **rank** data=sample descending ties=low;

ranks rankedni rankedtotalcomp;

var ni totalcomp;

**run**; **quit**;

This example will rank ni and totalcomp in descending order. The rank variable will be included in your dataset under the variable names rankedni and rankedtotalcomp. Thus the highest values of net income will be assigned a 0 and the highest values of total compensation will be assigned a 0. The default in SAS is to rank in ascending order, so if you wish to rank in descending you have to remember to include it in your code (as in this example). The ties option specifies how you wish ties to be ranked, the default is mean, or you can have SAS assign ties to the lowest rank of the group (low) or the highest rank of the group (high).

**Proc Surveyreg**

This procedure is an alternative to proc reg for fitting linear models. Proc reg makes the assumption that the researcher has drawn a truly random sample from the population and that their data meet all of the criteria for using OLS. However, the majority of the time in accounting or finance research we will not be drawing a truly random sample; therefore, we need a procedure that corrects the standard errors for this problem. The surveyreg procedure uses generalized least squares and its basic set up is the same as proc reg.

Example 1:

**proc** **surveyreg** data=sample;

model salary = age yrofeduc gender;

**run**; **quit**;

As with proc reg, the options are numerous and additional resources on this procedure can easily be found in various SAS guides and in online resources.

**Proc Univariate**

Univariate is another procedure in SAS that calculates descriptive statistics for a variable. In particular, proc univariate provides a rich description as to the distribution of a particular variable. The information provided is the mean, median, mode, standard deviation, skewness, and kurtosis along with tests for location, quantiles, and the extreme observations.

Example 1:

**proc** **univariate** data=sample;

var at ni csho;

**run**; **quit**;

The above example will return the litany of proc univariate descriptive statistics for the variables total assets, net income, and common shares outstanding.

Example 2:

**proc** **univariate** data = sample;

var at ni csho;

by sich;

**run**; **quit**;

By adding the by statement to the code the descriptive statistics will now be calculated for each of the three variables by SIC code.