A 602 SAS Primer

Introduction

SAS Version 9 is a powerful, very general purpose, 'high-level' programming language. It provides extensive capabilities to both process and analyze data. This document attempts to provide some initial structure so that a user who is unfamiliar with SAS can get started.

Overview

SAS processing of input is essentially divided into two steps: (1) the DATA step; and, (2) the PROC step. The DATA step reads in the names of variables, and the values for each observation on those variables. Data can be dropped, selected, and/or transformed in the DATA step. The PROC step represents one or more executions of a defined procedure intended to manipulate and/or analyze data.

Note that within a single execution of SAS it is possible to have both multiple DATA steps - with multiple sets of data - and/or, multiple PROC steps. Indeed, it is possible to have a SAS PROC create data which will become available for subsequent analyses.

The SAS language

SAS expects to process any of a variety of keywords that serve as commands. Many commands require the user provide additional information. All SAS commands end in a semi-colon!

All commands and user-supplied information can be entered either in upper- or lower-case. SAS does not (typically) distinguish between the two cases.

In the presentation below, the basic portion of a given SAS command is presented in all CAPITAL letters, while information to be provided by the user to the SAS command is in lower-case letters.

Alphameric means any combination of alphabetic and numeric characters, in which the first character is alphabetic.
The $DATA$ step:

At a minimum, the $DATA$ step requires 3 commands: $DATA$, $INPUT$, and $DATALINES$.

$DATA$ dataset ;

where

$DATA$ indicates to SAS the beginning of the $DATA$ step; and,

`dataset` is a name of 8 or less alphameric characters that SAS will use to create and keep track of a data set during its execution.

$INPUT$ varname1 ... varname_ ;

where

$INPUT$ tells SAS the names of each of the variables in the data set; and,

`varname1 ... varname_` are the names of the variables, each of which are 8 or less alphameric characters, separated by one or more blank spaces.

$DATALINES$ ;

where $DATALINES$ tell SAS that the data for each observation follow immediately. (A synonym for $DATALINES$ is $CARDS$.)

The data can be most easily input in what is known as free-field format, that is, with one or more blank spaces separating each value. For each observation, its `varname1` value comes first, then its `varname2` value, and so forth. The values for each observation begin on a new line.

The $PROC$ step:

There exist a variety of $PROC$s available for statistical analysis. Each procedure begins with the command $PROC$, followed by the name of the procedure. This initial command is often followed by any of a number of subcommands, which allow for user control of the $PROC$. 
SAS Graphics

SAS procedures CORR, REG, and GLM support Output Delivery System (ODS) graphics functionality in SAS 9.2. Therefore, these procedures will automatically produce high-quality graphics using SAS/GRAPH. Moreover, graphics output is also under control of the user.

In order to produce graphs and plots, (1) an output location for the graphs must be specified, and (2) ODS Graphics must be enabled:

ODS destination;
ODS GRAPHICS ON;
PROC name;
.
.
RUN;
QUIT;
ODS GRAPHICS OFF;
ODS destination CLOSE;

`destination` can be either HTML or RTF, which means that SAS will create an html or rtf file you can save and later edit;

`name` is CORR, REG, or GLM.

SAS Procedures

For our purposes, there are six basic PROCs we will use: PRINT, CORR, REG, GLM, PLOT, and CHART. A brief introduction to each of these PROCs follows.

**PROC PRINT:**

PRINT can be used to list data values for each observation, as:

PROC PRINT DATA=dataset options;
VAR varname1 ... varname_;

where

`dataset` is the name of the SAS dataset, created in a DATA step, or by a previously-executed PROC;
`options` include:

- **LABEL** uses the variable labels as the column headings;
- **N** prints a line at the end of the output giving the number of observations in the dataset; and,
- **NOOBS** omits printing of the observation number preceding each observation in the dataset.

`varname1 ... varname_` are the names of (one or more) variables, separated by one or more blank spaces, whose values are to be printed. (Omitting the VAR subcommand will produce a listing of all variables for all cases.)

**PROC CORR:**

**CORR** will provide Pearson product-moment correlations among a set of variables, together with some descriptive statistics and probability values, as:

```
PROC CORR DATA=dataset ;
VAR varname1 ... varname_ ;
```

where

`dataset` is the name of the SAS dataset, created in a **DATA** step, or by a previously-executed **PROC**; and,

`varname1 ... varname_` are the names of (two or more) variables, separated by one or more blank spaces, whose intercorrelations are to be determined. (Omitting the VAR subcommand will produce a correlation matrix of all variables.)

**PROC REG:**

**REG** is a very general-purpose ordinary least squares regression procedure, which provides a wide choice of analysis and output options. The basic approach would be:

```
PROC REG DATA=dataset poptions ;
mlabel: MODEL  crit = pred1 ... pred_ / moptons ;
tlabel: TEST effects ;
```

where
`dataset` is the name of the SAS dataset, created in a DATA step, or by a previously-executed PROC;

`poptions` include:

\textit{CORR} prints the intercorrelations of the variables in the \textit{MODEL} statement; and,

\textit{SIMPLE} prints various descriptive statistics for the variables in the \textit{MODEL} statement; and,

\textit{LINEPRINTER} specifies that the \textit{PLOT} subcommand (see below) produce low-resolution, dot-matrix-type scatterplots, rather than high-resolution scatterplots. [This option does not affect the \textit{PLOT=} option, which will always produce high-resolution graphics using ODS GRAPHICS.]

`mlabel` is a label of 8 or less alphameric characters that is printed on the output to identify the specific model being tested (useful when multiple \textit{MODEL} statements are used);

`crit` is the name of the criterion variable;

`pred1 ... pred_` are the names of the predictor variables, separated by one or more blank spaces;

`moptions` include:

\textit{SS2} prints the sums of squares associated with each predictor;

\textit{SCORR2} prints the squared semi-partial correlation coefficients;

\textit{PCORR2} prints the squared partial correlation coefficients;

\textit{STB} prints the standardized partial regression coefficients; and,

\textit{P} prints the predicted and residual value for each observation.

`tlabel` is a label of 8 or less alphameric characters that is printed on the output to identify the specific effect being tested (useful when multiple \textit{TEST} statements are used); and,

`effects` is the name of one or more predictor variables, followed by an equal sign, and a hypothesized value (for example: if X1 and X2 were predictors, \textit{TEST X1 = 0} would test the hypothesis that the regression coefficient associated with the predictor variable
X1 equals zero; or, \( TEST \ X1 - X2 = 0 \) would test the hypothesis that the regression coefficient associated with X1 equals the regression coefficient associated with X2.

**Note** that it is also possible to use REG to plot variables, simply by including the following subcommand some place after the `MODEL` subcommand:

\[
PLOT \ yvarname*xvarname='symbol' ;
\]

where

`yvarname` is the name of the variable to be plotted on the Y-axis;

`xvarname` is the name of the variable to be plotted on the X-axis; and,

``symbol`` is a single alphanemic character, enclosed in single quotes, with which each point will be represented in the plot.

**Note** that it is possible to include the observation number, predicted values, or residuals in a `PLOT` request by using the special `xvarname` or `yvarname` of `obs.`, `p.`, and/or `r.`, respectively.

**Note** that it is also possible to use REG to create a dataset containing the values of one or more variables, simply by including the following subcommand:

\[
OUTPUT OUT=regdata P=predname R=residname ;
\]

where

`regdata` is the name of a SAS data set containing all the variables in the input data set, plus the predicted values for each observation, `predname`, and the residual values for each observation, `residname`. This data set may then be referenced later, and its data printed, plotted, and so forth.

`predname` is an 8 character alphanemic variable name for the predicted values for each subject; and,

`residname` is an 8 character alphanemic variable name for the residual values for each subject.

**Note** that it is possible to use multiple `TEST` and `PLOT` subcommands in a single execution of `REG`.

**PROC GLM:**
GLM represents an exceedingly general implementation of the General Linear Model (hence its name), and thus, has both considerable power and flexibility. It will conduct, among other analyses, simple and multiple regression, analysis of variance, and analysis of covariance:

```
PROC GLM DATA=dataset options ;
CLASS factor(s) ;
MODEL dependent = effect(s) / doptions ;
RANDOM reffect(s) / roptions ;
TEST H=heffect E=eeffect ;
CONTRAST 'label' ceffect(s) coefficients ;
ESTIMATE 'label' ceffect(s) coefficients ;
LSMEANS leffect(s) / loptions ;
REPEATED weffect wlevels / woptions ;
```

where

`dataset` is the name of the SAS dataset, created in a `DATA` step, or by a previously-executed `PROC`;

`options` include:

- `ORDER=INTERNAL`, which indicates that the levels of the factor(s) are organized according to the numeric values read in the input data for the variables used to name the factor(s). [Otherwise, if `PROC FORMAT` is used, levels of the factor(s) are organized according to the alphameric order of the formatted values. `PROC FORMAT` is not covered in this primer; it is used in the example problem.]

`factors` are the names of (one or more) variables whose values represent levels of the factor(s);

`dependent` is the name of the dependent variable;

`effects` are the names of the variables that represent effects to be included in the model. These effects can include: **main effects** – specified as the names of the factors [for example: A]; **interactions** – specified with a asterisk between the factor names [for example: A*B]; and **nested effects** – specified as the name of the factor, followed immediately by parentheses around the name of the factor within which it is nested [for example: B(A), meaning B is nested within A].

`doptions` include:

- `NOUNI` suppresses printing of the univariate tests of significance (often useful when the `REPEATED` subcommand is used);
EFFECTSIZE produces effect size information, including the estimates for
the semipartial-$\omega^2$ and the partial-$\omega^2$ and estimates and confidence
intervals for the semipartial-$\eta^2$ and the partial-$\eta^2$ for each fixed effect and
specified CONTRASTs on fixed effects in the model;

SS3 prints the sums of squares and $F$ value for each effect in the model;
and,

$P$ prints the predicted and residual value for each observation.

`reffect(s)` is the name of one or more effects (see `effects` directly above) included in
the `MODEL` statement that will be considered to be random rather than fixed. [GLM will
consider all effects as fixed effects unless you use the RANDOM statement to
specifically designate the random effect(s).]

`roptions` include:

TEST requests tests of significance for the random effects in the model.

`heffect` is the name of an effect which will be tested against

`eeffect` is the name of an effect which will serve as the error term only for the `heffect` given on the TEST subcommand;

"label" is an alphabetic label of 20 or less characters, used on the output to identify the
contrast;

`ceffect(s)` is the name of one or more effects that appear in the `MODEL` statement,
upon which a contrast will be drawn, and/or an estimate of a mean difference will be
obtained;

`coefficients` are numbers that indicate exactly what means are to be compared;

`leffect(s)` is the name of one or more effects in the model for which least-squares
means are to be calculated;

`loptions` include:

STDERR prints the standard error of the least squares means and the $p$
values for the hypothesis that the least squares mean equals zero;

CL prints confidence limits for each of the least squares means;

$\text{ALPHA} = n$
where

`n` is a value between 0 and 1, and defines the alpha level for the confidence limits for the least squares means; and,

`PDIF` prints the p values for all possible comparisons of the means for the `leffect(s)`.

`ADJUST=amethod`

where

`amethod` can include either `BON`, `SCHFFE`, or `TUKEY`, and yields p values associated with the tests of the least squares means adjusted for the chosen multiple comparison procedure. [NOTE that the `PDIF` option must be specified along with `ADJUST`.]

`SLICE=sffect` or `SLICE=(seffects)`

where

`sffect` is the name of a fixed effect or effects by which the interaction `leffect(s)` are divided, yielding simple effects tests for the `leffect(s)` as a function of the `sffect`. [for example: if A*B is an effect in the model, then `SLICE=A` would test the test the simple main effects of B within each level of A, while `SLICE=(A B)` would test the simple main effects of B within each level of A, and the simple main effects of A within each level of B]. [Note that `ADJUST` does not adjust the p values for any tests associated with `SLICE`.]

`OUT=lsmeans`

where

`lsmeans` is the name of a SAS dataset containing the means of the dependent variable by the `leffect(s)` . This data set may then be referenced later, and its data printed, plotted, and so forth

`weffect` is the name to be given to a within-subject factor; and,

`woptions` include:

`NOM` suppresses printing of the multivariate tests of significance.
**Note** that it is also possible to use **GLM** to create a dataset containing the values of one or more variables, simply by including the following subcommand some place after the **MODEL** subcommand:

```
OUTPUT OUT=glmdata P=predname R=residname ;
```

where

`glmdata` is the name of a SAS data set containing all the variables in the input data set, plus the predicted values for each observation, `predname`, and the residual values for each observation, `residname`. This data set may then be referenced later, and its data printed, plotted, and so forth.

`predname` is an 8 character alphameric variable name for the predicted values for each subject; and,

`residname` is an 8 character alphameric variable name for the residual values for each subject.

**PROC PLOT:**

**PLOT** produces any of a variety of Euclidean (i.e., X-Y) plots, as:

```
PROC PLOT DATA=dataset ;
PLOT yvarname*xvarname / options ; and/or
PLOT yvarname*xvarname = 'symbol'/ options ; and/or
PLOT yvarname*xvarname = zvarname / options ;
```

where

`dataset` is the name of the SAS dataset, created in a **DATA** step, or by a previously-executed **PROC**; and,

`yvarname` is the name of the variable to be plotted on the Y-axis;

`xvarname` is the name of the variable to be plotted on the X-axis;

`'symbol'` is a single alphameric character, enclosed in single quotes, with which each point will be represented in the plot; and,

`zvarname` is the name of a variable whose values will mark each point in the plot; and,

`options` include:
OVERLAY indicates that all the plots requested with a PLOT subcommand are to be printed on a single page.

Note that it is possible to use multiple PLOT subcommands in a single execution of PLOT.

Note that PLOT produces low-resolution, dot-matrix-style plots. The SAS/GRAPH module contains PROC GPLOT, which produces the same plots as PLOT, in high-resolution, presentation-quality format. PROC GPLOT is extremely powerful and flexible, and control of the features of the plot are under direct control of the programmer.

PROC CHART:

CHART produces a variety of bar graphs and charts, both vertical and horizontal, as:

PROC CHART DATA=dataset ;
VBAR effect / options ;
HBAR effect / options ;

where

‘dataset’ is the name of the SAS dataset, created in a DATA step, or by a previously-executed PROC;
‘effect’ is the name of (one or more) variables whose values are to be plotted; and,
‘options’ include:

DISCRETE indicates the variable is discrete rather than continuous;

TYPE=MEAN indicates that each bar should be drawn at the mean of the varname specified in the SUMVAR subcommand;

SUMVAR=varname

where

‘varname’ is the name of the variable to be plotted in the bar chart;

MIDPOINTS=values
where

`values` define midpoints of the range of values for the chart variable over
which each bar will be drawn;

`REF=value`

where

`value` is a number, at which point a reference line will be drawn on the
response axis.

**Note** that it is possible to use multiple `VBAR` and/or `HBAR` subcommands in a
single execution of `CHART`.

**Note** that `CHART` produces low-resolution, dot-matrix-style charts and graphs.
The SAS/GRAPH module contains `PROC GCHART`, which produces the same charts
and graphs as `CHART`, in high-resolution, presentation-quality format. `PROC GCHART`
is extremely powerful and flexible, and control of the features of the chart are under
direct control of the programmer.
In closing, please recognize that this introduction only presents the barest of essentials. Hence, many of the capabilities of a given PROC are ignored. And, there are better (where better means both faster and more efficient) ways of accomplishing many of the tasks required throughout the semester. However, what’s presented herein will get you started – and will also get you finished – without a degree in statistics or computer science.

Want to know even more? SAS is very glad you asked. At last count, the SAS Institute, marketers of SAS, was selling in excess of 100 different manuals for the SAS System. Perhaps the first several to purchase might be:

- *SAS Introductory Guide*
- *SAS Language: Reference*
- *SAS Language and Procedures: Introduction*
- *SAS Language and Procedures: Usage 1 and 2*
- *SAS Procedures Guide*
- *SAS/STAT User’s Guide, Volume 1 and 2*

All are typically on sale in the UMCP Bookstore.

Want it all? Go to: www.sas.com