Introduction to SAS Procedures

Sarah Bell

Overview

- Review
- Basic syntax
- Procedures
- Elements of style
- Data manipulation
- Basic statistics

SAS Procedures SAS/STAT

- ACECLUS
- ANOVA
- BOXPLOT
- CALIS
- CANCORR
- CATMOD
- CLUSTER
- CORRESP
- DISCRIM
- DISTANCE
- FACTOR
- FASTCLUS
- FREQ
- GAM
- GENMOD
- GLIMMIX
- GLM
- GLMMOD
- GLMPOWER
- GLMSELECT
- HPMIXED
- INBREED
- KDE
- KRIGE2D
- LATTICE
- LIFEREG
- LIFETEST
- LOESS
- LOGISTIC
- MCMC
- MDS
- MI
- MIANALYZE
- MIXED
- MODECLUS
- MULTTEST
- NESTED
- NLIN
- NLMIXED
- NPAR1WAY
- ORTHOREG
- PHREG
- PLAN
- PLM
- PLS
- POWER
- PRINCOMP
- PRINQUAL
- PROBIT
- QUANTREG
- REG
- ROBUSTREG
- RSREG
- SCORE
- SEQDESIGN
- SEQTEST
- SIM2D
- SMNORMAL
- STDIZE
- STEPDISC
- SURVEYFREQ
- SURVEYLOGISITIC
- SURVEYMEANS
- SURVEYPHREG
- SURVEYREG
- SURVEYSELECT
- TPSPLINE
- TRANSREG
- TREE
- TTEST
- VARCLUS
- VARCOMP
- VARIOGRAM

PROC Step

- Each procedure (PROC) has unique characteristics
- Basic PROC structure is similar to:

```
proc __________ data= ___________
      <other proc-specific options>;
      by _____________________________;
      <proc-specific statement(s)>;
      label ___________________________;
      format __________________________;
run; <and/or> quit;
```

PROC PRINT

- Used to organize and display data in the ‘output’ window
- Has many options to control the appearance of data
- Mainly lists data, but has some selection, grouping, and summary capabilities

PROC PRINT

```
proc print data=dataset <options>;
      by <descending> variable-1…<descending>
      variable-n <notsorted>;
            pageby by-variable;
            sumby by-variable;
      id variables <options>;
      sum variables <options>;
      var variables <options>;
run;
```

PROC CONTENTS

- Shows the contents of one or more SAS datasets
  - Default output orders variables alphabetically by name
  - Use VARNUM to list by column position
  - Can output ‘metadata’
- Prints the directory of the SAS library

PROC CONTENTS

```
proc contents data=dataset <options>;
run;
```

PROC SORT

- Used to organized datasets typically in preparation for ‘by’ processing
- Can be ascending or descending
- Can include one to all the variables in a dataset
- Can create new datasets
- Can be used to eliminate duplication

PROC SORT

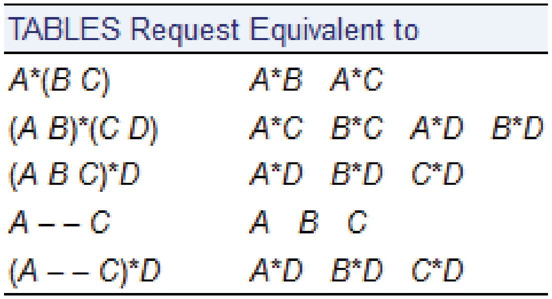
```
proc sort data=dataset <options>;
     by <descending> variable-1…
     <descending> variable-n;
run;
```

PROC FREQ

- Useful for examining categorical variables
- Reports counts and percentages
- If ‘by’ variable is specified, data must be pre-sorted

PROC FREQ

- Tables can be crossed



TABLES Request	Equivalent to
A*(B C)	A*B A*C
(A B)*(C D)	A*C B*C A*D B*D
(A B C)*D	A*D B*D C*D
A - - C	A B C
(A - - C)*D	A*D B*D C*D

PROC FREQ

```
proc freq data=dataset <options>;
     by variables;
     exact statistic-options </options>;
     output <out=dataset> options;
     tables requests </options>;
     test options;
     weight variable </option>;
run;
```

PROC MEANS

- Used for descriptive statistics of numerical variables
- If ‘by’ variable is specified, data must be pre-sorted
- Alternatively, the ‘class’ statement can be used to report by categories in other variables

PROC MEANS

```
proc means data=dataset <options> <statistic-keywords>;
      by variables;
      class variables </options>;
      freq variable;
      id variables;
      output <out=dataset> options;
      types request(s);
      var variables;
      ways list;
      weight variable;
run;
```

PROC UNIVARIATE

- Use for descriptive statistics of numerical variables
- If ‘by’ variable is specified, data must be pre-sorted
- Alternatively, the ‘class’ statement can be used to report by categories in other variables

PROC UNIVARIATE

```
proc univariate data=dataset <options>;
      by variables;
      class variables <v-options>;
      freq variable;
      histogram variables </options>;
      id variables;
      output <out=dataset> options;
      qqplot variables </options>;
      var variables;
      weight variable;
run;
```

Break

Elements of Style

```
data trial1;infile 'C:\wagedata.txt'; input id days wages;wage_rate
=wages/days;if wage_rate>20 then lvl='hi';else lvl='lo';run;
```

```
data trial1;  Infile 'C:\wagedata.txt';
Input                                id
days                             wages;
wage_rate                            =
wages/                            days;
if                         wage_rate>20
then lvl='hi';   else   lvl='lo';  run;
```

```
data
trial1;
Infile
'C:\
wagedata.txt
';
input
id
days
wages;
wage_rate
=
wages/
days;
if
wage_rate>20
then
lvl=
'hi';
else
lvl='lo';
run;
```

Elements of Style

- Large block comment describing the program and purpose
- Include comments before important DATA and PROC steps
- One statement per line
- Insert a blank line before each DATA or PROC step
- Left-justify all DATA, PROC, and RUN statements. Indent all statements within a DATA or PROC step

Elements of Style

```
/* this is a sample program used to demonstrate some
of the basic elements of programming style  */

data trial1;
        infile 'C:\wagedata.txt';
        input id days wages;
        wage_rate=wages/days;

        * “20” is industry standard for hi;
        if wage_rate>20 then lvl='hi';
        else lvl='lo';
run;
```

Large block comment at beginning describing program and purpose

One statement per line

Blank line to separate sections of the program

Short comment to explain code

Indenting subordinate statements

PROC TRANSPOSE

- Flips data on its side
- Recommended:
  - Do in small chunks
  - Compare original and transposed dataset
- With experience you can transpose multiple variables simultaneously

Merging

- The MERGE statement is used to combine two or more SAS datasets
- Can be merged by a ‘key’ variable, or a group of variables that create a unique key
  - Many types of merges
  - 8 different ways to do a simple merge in SAS

Merging

Patient Data

patno	lname	sex
11	Jones	M
66	Smith	M
33	Brown	F
55	Harris	F
44	Anderson	F
22	Collins	M

Visit Data

patno	visit #	wt
11	1	137
11	2	135
33	1	186
33	2	182
33	3	176
66	1	157

“Merged” Data

patno	lname	sex	visit #	wt
11	Jones	M	1	137
11	Jones	M	2	135
22	Collins	M	.	.
33	Brown	F	1	186
33	Brown	F	2	182
33	Brown	F	3	176
44	Anderson	F	.	.
55	Harris	F	.	.
66	Smith	M	1	157

PROC SQL

Structured Query Language (SQL)
A language used for managing data in many different computer applications (primarily database applications). It has been available in SAS since the late 1980's and can be used for a wide variety of purposes including nearly everything we have done this morning.

Syntax:

```
PROC SQL;
  SQL statements;
quit;
```

Chi-Square

- Used to examine the association between two categorical variables
- Used to determine if the distribution of one categorical variable is different across the levels of a second categorical variable

Chi-Square

```
proc freq data=data;
     tables CategoricalVariable *
     CategoricalVariable / chisq;
run;
```

T-Test

- One-sample
  - Used to examine whether the sample mean of a single continuous variable in a single group of individuals is different from a hypothesized population value

T-Test

- One-sample

```
proc ttest data=data
     h0=HypothesizedValue;
     var ContinuousVariable;
run;
```

T-Test

- Two-Sample
  - Used to examine whether the sample mean of a continuous variable is different between two independent groups

T-Test

- Two-Sample

```
proc ttest data=data;
     class GroupVariable;
     var ContinuousVariable;
run;
```

T-Test

- Paired
  - Used to compare two sample means when the samples are not independent
  - Examples:
    - Pre- and post-test scores
    - Case-control comparison

T-Test

- Paired

```
proc ttest data=data;
     paired ContinuousVariable *
     ContinuousVariable;
run;
```

Correlation

- Used to determine whether and how strongly two continuous or ordinal variables are related

Correlation

```
proc corr data=data;
     var ContinuousVariables;
run;
```

ANOVA

- Used to examine whether the sample mean of a continuous variable is different between two or more groups

ANOVA

- Best used when design is well balanced

```
proc anova data=data;
     class CategoricalVariable;
     model ContinuousVariable =
     CategoricalVariable;
run;
```

Simple Linear Regression

- Used to fit a single line through a scatterplot
- Regression estimates used to explain the relationship between one independent variable and one dependent variable.

Simple Linear Regression

- Doesn’t support ‘class’ statement

```
proc reg data=data;
     model ContinuousVariable =
     ContinuousVariable or
     IndicatorVariable;
run;
```

Simple Linear Regression

```
proc glm data=data;
     model ContinuousVariable =
     ContinuousVariable;
run;
proc glm data=data;
     class CategoricalVariable;
     model ContinuousVariable =
     CategoricalVariable;
run;
```

Survival Curve

- Statistical picture of the survival experience of a group of individuals

Survival Curve

```
proc lifetest data=data;
     time FollowUpTime *
     CensoringVariable
     (CensoringValue);
     strata GroupVariable;
run;
```