SAS/STAT Procedures Reference Sheet

Overview of SAS/STAT

This software provides comprehensive statistical tools for a variety of statistical analyses. There are over 90 procedures that can be used for statistical analysis using SAS/STAT. This reference sheet, however, will focus on the fundamentals and provide guidance for employing commonly used procedures.

General Syntax Conventions

Consider the following statements as demonstrations of general syntax rules when navigating the reference sheet:

MODEL *response <(response-options)> = <fixed-effects> </ model-options>*;
CONTRAST *'label' contrast-specification <, …> = <fixed-effects> </ options>*;

Syntax	Guidelines
UPPERCASE BOLD	used for statement names/keywords
oblique	represent arguments for which you supply a value
< >	identify optional arguments
…	ellipsis dots indicate that the preceding argument can be repeated
( )	parentheses indicate arguments that must be grouped together
|	a vertical bar indicates that you can choose one value from a group of values
;	semicolon indicates the end of a statement
= ~ : /	other special characters indicate where in the syntax you must type those characters

Procedures for Data Exploration

PROC SGPLOT: creates single-cell plots along with a variety of plot and chart types

- Can generate scatterplots, box and whisker, histograms (portrayed below), series plots, needle plots

PROC SGPLOT DATA = *data-set-name <options>;*
  *<**BY** variable;>*
  *<**WHERE** expression;>*
  *<**HISTOGRAM** response-variable </options>;*
RUN;

PROC MEANS: generates simple summary statistics for each numeric column in the input data by default unless the VAR statement is used

- **CLASS** specifies variables to group data before calculating statistics
- **WAYS** specifies number of ways to make unique combinations of class variables
- **OUTPUT** provides the option to create an output table and specific output statistics
- **OUT =** names the output table to be created

PROC MEANS DATA = *data-set-name;*
  *<**WHERE** expression;>*
  *<**VAR** col-name(s);>*
  *<**CLASS** col-names(s);>*
  *<**WAYS** n;>*
  *<**OUTPUT OUT =** output-table <statistic =col-name>*
RUN;

PROC UNIVARIATE: generates summary statistics and more detailed statistics about distribution and extreme values for each numeric variable by default

PROC UNIVARIATE DATA = *data-set-name;*
  *<**VAR** col-name(s);>*
  *<**WHERE** expression;>*
RUN;

PROC CORR: computes correlation coefficients to determine strength and significance of linear relationships

- **BY** statement prompts separate analyses of specified group data
- **FREQ** statement lists a numeric variable to represent frequency of the observation
- **ID** statement specifies additional variables to identify observations in scatterplots
- **PARTIAL** statement identifies controlling variables to compute Pearson, Spearman, or Kendall partial-correlation coefficients
- **VAR** statement list variables for which to compute correlation coefficients
- **WITH** statement lists variables to compute correlations for that are included in the VAR statement

PROC CORR <options>;
  <**BY** *variables*;>
  <**FREQ** *variable*;>
  <**ID** *variables*;>
  <**PARTIAL** *variables*;>
  <**VAR** *variables*;>
  <**WEIGHT** *variable*;>
  <**WITH** *variables*;>
RUN;

Procedures for Data Management

PROC RANK: computes ranks for one or more numeric variables across the observations of a SAS dataset and writes the ranks to a new dataset

- **BY** calculates a separate set of ranks for each BY group
- **VAR** specifies the variables to rank
- **RANKS** identifies a variable to which the rank are assigned

PROC RANK <options>;
  BY <DESCENDING**>** *variable – 1*
    <<DESCENDING> *variable – 2 …>*
    <NOTSORTED>*;*
  VAR *data-set-variables*;>
  RANKS *new-variables*;
RUN;

* Note: NOTSORTED specifies that the observations are not necessarily sorted in alphabetic or numeric order

SAS/STAT Procedures Reference Sheet

PROC CLUSTER: hierarchically clusters the observations in a SAS dataset

- **BY** can be used to obtain separate analyses of observations in groups that are defined by the BY variables
- **COPY** copies the variables in this statement from the input dataset to the OUTTREE = dataset
- **FREQ** used if one variable in the input dataset represents the frequency of occurrence for other values in the observation
- **ID** identifies observations in the displayed cluster history and in the OUTTREE = dataset
- **RMSSTD** if the coordinates in the DATA = dataset represent cluster means, this statement specifies the variable containing root mean squared standard deviations
- **VAR** lists numeric variables to be used in the cluster analysis

* Note: METHOD = specifies clustering method

```
PROC CLUSTER METHOD =name <options>;
        BY variables;
        <COPY variables;>
        FREQ variables;
        <ID variable;>
        RMSSTD variable;
        <VAR variables;>
RUN;
```

PROC SURVEYSELECT: provides a variety of methods for selecting probability-based random samples

- **CONTROL** names variable(s) for sort the input dataset before sample selection
- **FREQ** names a numeric variable that contains frequency of occurrence of each observation
- **ID** names variable(s) from the DATA = input dataset to include in the OUT = dataset of selected units
- **SAMPLINGUNIT | CLUSTER** variable(s) that identify the sampling units as groups of observations (clusters)
- **SIZE** variable(s) that contains size measures that are used for PPS selection (must be numeric)
- **STRATA** obtains stratified sampling

```
PROC SURVYSELECT <options>;
        CONTROL variables;
        FREQ variable;
        ID variables;
        SAMPLINGUNIT | CLUSTER variables </options>;
        SIZE variable;
        STRATA variables </options>;
RUN;
```

Procedures for Continuous Response Analysis

PROC TTEST: performs t tests and computes confidence limits for one-sample, paired observations, independent samples, and AB/BA crossover design

- **BOOTSTRAP** requests bootstrap standard error, bias estimates, and confidence intervals
- **CLASS** gives the name of classification (grouping) variable
- **PAIRED** identifies variables to be compared in paired ttest
- **BY** used to obtain separates analyses of observations in groups
- **VAR** names variables to be used in the analyses
- **FREQ** identifies variables that contain the frequency for each observation
- **WEIGHT** weights each observation in the input dataset by the specified variable

```
PROC TTEST <options>;
        BOOTSTRAP variables </options>;
        CLASS variable;
        PAIRED variables;
        BY variables;
        VAR variable </options>;
        FREQ variables;
        WEIGHT variable;
RUN;
```

PROC REG: general-purpose procedure for conducting regression analysis

- **MODEL** specifies the dependent and independent variables in the regression model, requests a model selection method, displays predicted values, and provides details on the estimates (according to which options are selected)
- **ADD / DELETE** adds or deletes independent variables to the regression model
- **BY** specifies variables to define subgroups for the analysis
- **FREQ** specifies a frequency variable
- **ID** names a variable to identify observations in the tables
- **MTEST** performs multivariate tests across multiple dependent variables.
- **OUTPUT** creates an output data set and names the variables to contain predicted values, residuals, and other diagnostic statistics
- **PLOT** generates scatter plot

```
PROC REG <options>;
        <label:> MODEL dependents = <regressors> </ options>;
        <BY variables;>
        <FREQ variable;>
        <ID variables;>
        <VAR variables;>
        <ADD variables;>
        <DELETE variables;>
        <label:> MTEST <equation, …, equation> </ options>;
        <PLOT <yvariable*xvariable> <=symbol>
            <…yvariable*xvariable> <=symbol> </ options>;>
RUN;
```

PROC GLM: uses the method of least squares to fit general linear models including regression, ANOVA, ANCOVA, MANOVA, partial correlation

- **MODEL** defines the model to be fit
- **ABSORB** absorbs classification effects in a model
- **CLASS d**eclares classification variables
- **CONTRAST** constructs and tests linear functions of the parameters
- **ESTIMATE** estimates linear functions of the parameters
- **ID** identifies observations on output
- **LSMEANS** computes least squares (marginal) means
- **MANOVA** performs a multivariate analysis of variance
- **MEANS** computes and optionally compares arithmetic means
- **OUTPUT** requests an output data set containing diagnostics for each observation
- **RANDOM** declares certain effects to be random and computes expected mean squares
- **REPEATED** performs multivariate and univariate repeated measures analysis of variance
- **TEST** constructs tests that use the sums of squares for effects and the error term you specify
- **STORE** saves the context and results of the statistical analysis
- **WEIGHT** specifies a variable for weighting observations

SAS/STAT Procedures Reference Sheet

```
PROC GLM <options>;
    MODEL dependent-variables = independent-effects </
        options>;
    <ABSORB variables;>
    <CLASS variable <(REF= option)> …<variable <(REF=
        option)>> </ global-options>;>
    <CONTRAST 'label' effect values <…effect values> </
        options>;>
    <BY variables;>
    <FREQ variable;>
    <ID variables;>
    <WEIGHT variable;>
    <ESTIMATE 'label' effect values <…effect values> </
        options>;>
    <LSMEANS effects </ options>;>
    <MANOVA <test-options> </ detail-options>;>
    <MEANS effects </ options>;>
    <OUTPUT <OUT=SAS-data-set> keyword=names
        <…keyword=names> </ option>;>
    <RANDOM effects </ options>;>
    <REPEATED factor-specification </ options>;>
    <TEST <H=effects> E=effect </ options>;>
    <STORE <OUT=>item-store-name </ LABEL='label'>;>
RUN;
```

PROC GLMSELECT: performs effect selection in the framework of general linear models. Similar to REG and GLM procedures; supports a variety of model-selection methods but does not support a CLASS statement

- **EFFECT** enables construction of columns for design matrices
- **MODELAVERAGE** requests that model selection be repeated on resampled subsets of the input data
- **PARTITION** specifies how observations are partitioned for model training, validation, and testing
- **SCORE** creates a new SAS data set containing predicted values and optionally residuals for data
- **STORE** saves the context and results of the statistical analysis

```
PROC GLMSELECT <options>;
    MODEL dependent-variables = independent-effects
        </options>;
    <BY variables;>
    <FREQ variable;>
    <WEIGHT variable;>
    <MODELAVERAGE <options>;>
    <PARTITION <options>;>
    <SCORE <DATA=SAS-data-set> <OUT=SAS-data-set>;>
    <STORE <OUT=>item-store-name </ LABEL='label'>;>
    <EFFECT name = effect-type (variables </ options>);>
RUN;
```

PROC PLM: performs post-fitting statistical analyses for the content of a SAS item store that was previously created with the STORE statement

- **EFFECTPLOT** produces a display of the fitted model
- **LSMESTIMATE** obtains custom hypothesis tests among least squares means
- **SHOW** uses the Output Delivery System (ODS) to display contents of the item store
- **SLICE** performs a partitioned analysis of the LS-means for an interaction
- **FILTER** enables you to filter the results of the PLM procedure

```
PROC PLM RESTORE=item-store-specification <options>;
    <FILTER
    <EFFECTPLOT <plot-type <(plot-definition-options)>>
        </options>;>
    <ESTIMATE <'label'> estimate-specification <(divisor=n)><,
        …<'label'> estimate-specification <(divisor=n)>>
        </options>;
    <LSMESTIMATE model-effect <'label'> values
        <divisor=n><, …<'label'> values <divisor=n>> </
        options>;>
    <SCORE DATA=SAS-data-set <OUT=SAS-data-
        set><keyword<=name>> …<keyword<=name>>
        </options>;>
    <SHOW options;
    <SLICE model-effect </ options>;>
    <TEST <model-effects> </ options>;>
RUN;
```

Procedures for Categorical Data Analysis

PROC FREQ: produces contingency (crosstabulation) tables and can computes various statistics to examine relationships between two classification variables

- **BY** provides separate analyses for each BY group
- **EXACT** requests exact tests
- **TABLES** specifies tables and requests analyses
- **OUTPUT** requests an output dataset
- **TEST** requests tests for measure of association and agreement
- **WEIGHT** identifies a weight variable

```
PROC FREQ <options>;
    <BY variables;>
    <EXACT statistic-options </computation-options>;
    <OUTPUT <OUT = SAS-dataset> output-options;
    <TABLES requests </ options>;
    <TEST options;>
    <WEIGHT variable </ option>;>
RUN;
```

PROC LOGISTIC: fits linear logistic regression models for discrete response data with maximum-likelihood methods

- **MODEL** names the response variable and the explanatory effects, including covariates, main effects, interactions, and nested effects
- **BY** provides separate analyses for each BY group
- **CLASS** names the classification variables to be used as explanatory variables in the analysis
- **EXACT** performs exact tests of the parameters for the specified effects
- **EXACTOPTIONS** specifies options that apply to every EXACT statement in the program
- **CONTRAST** provides a mechanism for obtaining customized hypothesis tests
- **EFFECTPLOT** produces a display of the fitted model
- **NLOPTIONS** controls the optimization process for conditional analyses and for partial slope models
- **ODDSRATIO** produces odds ratios for a variable even when the variable is involved in interactions with other covariates, and for classification variables that use any parameterization

SAS/STAT Procedures Reference Sheet

- **ROC** specify models to be used in the ROC comparisons
- **ROCCONTRAST** compares the different ROC models
- **UNITS** enables you to specify units of change for the continuous explanatory variables so that customized odds ratios can be estimated
- **WEIGHT** weights each observation in the dataset by the specified WEIGHT variable

```
PROC LOGISTIC <options>;
    <label:> MODEL variable <(variable_options)> =
        <effects> </options>;
    <BY variables;>
    <CLASS variable <(options)> <variable
        (options)>…</options>;
    <EXACT <'label'> <INTERCEPT> <effects> </options>;
    <EXACTOPTIONS options;>
    <CONTRAST 'label' effect values <, effect values, …>
        </options>;>
    <EFFECTPLOT <plot-type <(plot-definition-options)>> </
        options>;>
    <NLOPTIONS options;>
    <ODDSRATIO <'label'> variable </options>;>
    <ROC <'label'> <specification> </options>;>
    <ROCCONTRAST <'label'> variable </options>;>
    <UNITS <independent1 = list1 <independent2 = list2>>
        </options>;>
    <WEIGHT variable </options>;>
RUN;
```

Additional SAS Syntax

Often, analytical finders and models need to be delivered using a report. The following procedures and options provide functionality to share results.

Exporting Data

- Export a SAS dataset to variable file types (XLSX, TXT,CSV, etc) using a **PROC EXPORT** step
  - PROC EXPORT must be used to export unstructured data type. (e.g. CSV files)
  - **DMBS =** the database management system which specifies the type of data to export. (e.g. CSV, DLM, JMP, TAB)

```
PROC EXPORT DATA=input-table
                OUTFILE="output-file"
                <DBMS = identifier REPLACE> ;
RUN;
```

- Alternatively use a LIBNAME statement to export data.
- A LIBNAME statement can only be used if the output data type has an accessible SAS Engine (e.g. XLSX, JSON, XML).
- Ensure a LIBNAME libref CLEAR statement is used at the end to close the connection to the excel workbook.

```
LIBNAME myXL XLSX "C:/documents/Shopping.XLSX";
DATA myXL.shopping;
    SET work.shopping;
RUN;
LIBNAME myXL CLEAR
```

Exporting Reports

- The SAS Output Delivery System (ODS) can send reports to various file types to display reports including CSV, PowerPoint, RTF, and PDF.
- Each output type holds the same basic structure to open and close a file. Additional statements are available and optionable based on the file type.

```
ODS <destination> < destination specifications>;
        /* SAS Code that produces output */
ODS destination CLOSE;
```

- Additional options to excel files include:
  - Adding a style
  - Adding a worksheet label

```
ODS EXCEL FILE="filename.xlsx"
        STYLE=style
        OPTIONS(SHEET_NAME='label') ;
        /* SAS code that produces output on first
           worksheet */
        ODS EXCEL OPTIONS(SHEET_NAME='label');
        /* SAS code that    produces output on second
        worksheet */
ODS EXCEL CLOSE;
```

- PDF outputs can include a Table of Contents (PDFTOC) and Procedure labels in the bookmarks.

```
ODS PDF FILE="filename.xlsx"
        STYLE=style
        STARTPAGE = NO PDFTOC= 1;
        ODS PROCLABEL "label";
        /* SAS code that produces output */
ODS PDF CLOSE;
```

Exporting Reports

- You can use the ODS GRAPHICS statement options to control many aspects of your graphics. The settings that you specify remain in effect for all graphics in the current session until you change or reset these settings with another ODS GRAPHICS statement.

```
ODS GRAPHICS < OFF | ON> </ options>;
```

Additional Information

- For more information on SAS programming techniques, visit go.documentation.sas.com