SAS is a statistical software package that allows the user to manipulate and analyze data in many different ways. It includes four windows --- Program editor, Explorer, Log, and Output window. Log window is very useful, which inform you of any errors in your program and the reason for the errors. The main interface of SAS is command-line, which allows users better control the data. And some procedures (such as import data & power analysis) can also be executed by pull-down window. SAS runs on a wider variety of computers and operating system. Because of these advantages, this software package is used in many disciplines (not just statistics!), including medical sciences, biological sciences, and social sciences.

The SAS system provides many tools for creating statistical graphics. In many cases, multiple tools can be used to make a graph. SAS could create customized stand alone graphs and you could save and edit graphs like Word for papers and presentation. Besides, SAS could produce 3-D plots and maps. Several examples are given below. For more information, please refer book “Statistical Graphics in SAS” by Warren.

### Linear regression modeling

**Proc reg**
It is a procedure for regression and allows multiple model statements in one procedure. It can do model selection, and even plots summary statistics and normal QQ-plots. You can specify several PLOT statements for each MODEL statement, and you can specify more than one plot in each PLOT statement.

```sas
proc reg data=data_name;
    model weight=age;
    plot weight*age;plot predicted.*age;
    plot residual.*age;
    plot nqq.*residual.; run;
```

In the above example, a simple regression is performed with weight as the response and age as the explanatory variable. The plot statements request four plots: weight versus age, predicted values of weight versus age, residuals versus age, and normal QQ-plot versus residuals. predicted., residual., and nqq. are keywords that SAS recognizes.

**Proc glm**

It performs simple and multiple regressions, analysis of variance (ANOVA), analysis of covariance, multivariate analysis of variance, and repeated measures analysis of variance.

```sas
proc glm data=data_name;
    model weight=age; output out=a3 p=pred r=resid; run;
```

The example above performs a simple linear regression with weight as the dependent variable and age the independent variable. The predicted values of weight (the dependent variable) and the residuals are saved in a new dataset called a3 using the output statement. For multiple regression, simply list in the model statement all the variables on the right hand side of the equal sign with one space in between, e.g. model weight=age height;

In the case of ANOVA, a class statement is needed for categorical variables before the model statement. The example shows an ANOVA analyzing the effect of gender on weight.

```sas
proc glm data=a1;
    class gender; model weight=gender; run;
```

**Non-linear modeling**

**Proc nlin**

A regression model is called nonlinear, if the derivatives of the model with respect to the model parameters depend on one or more parameters. We wish to fit the log-logistic model \( y = \delta + (\alpha - \delta) / (1 + \exp(\beta \log(x/\gamma))) + e \) to the data. In this example, there is a rate variable and a response variable y.

```sas
proc nlin data=weeds;
    parameters alpha=100 delta=4 beta=2.0 gamma=0.2;
    model y = delta + (alpha-delta) / (1 + exp(beta*log(rate/gamma))); run;
```

The PARAMETERS statement defines which elements of the model statement are parameters and to be estimated. MODEL statement contains the mathematical expression of the model, apart from the error term. The only variable on the left hand side not defined in the PARAMETERS statement is RATE, which is looked for in the data set WEEDS.

**Proc nlmixed**

The next example is a count model using a poisson distribution. The poisson regression model uses response variable award which is a count of the number of awards received. For more information please refer “http://support.sas.com/rnd/app/papers/nlmixedsugi.pdf”.

```sas
proc nlmixed data='D:\data\hsbdemo.sas7bdat';
```
parms b0=0 b1=0 b2=0; xb=b0+b1*read+b2*female; mu = exp(xb);
model awards ~ poisson(mu); run;

Nested variable modeling

**Proc glm, Proc mixed, and Proc nested**

These three procedures could provide similar analyzes but PROC NESTED is easier to use and computationally more efficient for this special type of design. This is especially true when the design involves a large number of factors, levels, or observations. However, PROC NESTED assumes that the input data set is sorted by the classification (CLASS) variables.

```
proc glm
   class a b c d;
   model y=a b(a) c(a b) d(a b c);
```

```
proc nested
   class a b c d;
   var y;
```

**Continuous/ Categorical variable analysis**

**Proc freq**

The procedure allows producing frequency tables. Below, we make frequency tables for variable make.

```
PROC FREQ DATA=auto;
   TABLES make;
RUN;
```

The following example gives Chi-squared test.

```
proc freq data=ability;
   tables gender*results/chisq; weight count;
RUN;
```

**Proc means**

The procedure can be used to produce summary statistics. Below, we make descriptive statistics for the variable mpg.

```
PROC MEANS DATA=auto;
   CLASS foreign; VAR mpg;
RUN;
```

**Proc univariate**

The procedure could give get more detailed summary statistics.

```
PROC UNIVARIATE DATA=auto;
   VAR mpg;
RUN
```

**Proc catmod/ genmod/ glimmix/ logistic/ probit/ survey logistic**

These procedures used to fit the usual logistic regression model when there is a dichotomous/ ordinal/nominal response variable.

**Histogram**

Histograms can be produced using PROC UNIVARIATE/ CHART/GCHART. The example below demonstrates the use of PROC UNIVARIATE with the HISTOGRAM statement.

```
PROC UNIVARIATE DATA = sashelp.class NOPRINT;
   TITLE 'Summary of Weight Variable (in pounds)';
   HISTOGRAM weight / NORMAL; RUN;
```

**QQ-Plots**
QQ-plot could be produced under PROC UNIVARIATE/REG/CAPABILITY. The following statements create a Q-Q plot for variable Length. NORMAL option requests a normal line. MU= and SIGMA= normal-options request a distribution reference line with intercept 10 and slope 0.3 for each plot, corresponding to a normal distribution with mean \( \mu = 10 \) and standard deviation \( \sigma = 0.3 \). The SQUARE option displays the plot in a square frame, and the CTEXT= option specifies the text color.

```
proc univariate data=measures;
  qqplot length1 length2 / normal(mu=10 sigma=0.3) square ctext=blue;
run;
```

**Reading in multiple data formats**

Use the Import facility by clicking File menu – Import data. It starts a wizard that takes you through the process of converting the *.xls/ *.csv/ *.sav/ *.txt/ *.wk/*.dta/*.db into SAS. When the Import wizard asks for a Member, enter the SAS data set name you wish to assign to the data.

**Data manipulation** (e.g. log transformation in the data step)

```
data name; set name;
  lny   = log(variablename);     /* The natural logarithm (base e) */
  logy  = log10(variablename);   /* The log to base 10             */
run;
```

Please refer “http://www.ats.ucla.edu/stat/sas/library/SASTranMan_os.html#Transformations” for more information regarding data manipulation and transformation.

**PCA/Factor analysis**

**Proc factor**
The general form to perform a principal component analysis is presented below.

```
PROC FACTOR DATA=data-set-name
   METHOD=PRIN
   PRIORS=ONE
   MINEIGEN=p
   NFACT=n
   SCREE
   ROTATE=VARIMAX
   FLAG=desired-size-of-"significant"-factor-loadings;
   VAR variables-to-be-analyzed;
RUN;
```

Many options may be requested in factor statement. FLAG causes the printer to flag (with an asterisk) any factor loading whose absolute value is greater than some specified size. For example, if you specify FLAG=.35 an asterisk will appear next to any loading whose absolute value exceeds .35. Negative values are not allowed in the FLAG option, and the FLAG option should be used in conjunction with the ROUND option. METHOD specifies the method to be used in extracting the factors or components. The current program specifies principal factors method be used. MINEIGEN=p specifies the critical eigenvalue a component must display For example, the current program specifies MINEIGEN=1. This statement will cause PROC FACTOR to retain and rotate any component whose eigenvalue is 1.00 or larger. NFACT=n allows you to specify the number of components to be retained and rotated. PRIORS specify prior communality estimates. Users should always specify PRIORS=ONE to perform a principal component analysis. ROTATE specifies the rotation method. The preceding program requests a varimax rotation. Oblique rotations may also be requested. SCREE creates a plot that graphically displays the size of the eigenvalue associated with each component.
Random effects model

**Proc glm & Proc mixed**

Use RANDOM to contain all of the effects (or interactions) that we want to treat as random. In this case we have only put strain there.

```
PROC GLM DATA=ksh;
   CLASS strain site;
   MODEL activity = strain site strain*site;
   RANDOM strain;
RUN;
```

Sample size calculation

**Proc power**

The example below shows how to calculate the sample size for a given statistical power of testing the difference in the effect of two groups. We first specify the two means for both groups. In this case, the expected difference mean in two groups is 10. Next, we need specify the pooled deviation 16.03 and set the significant level 0.8.

```
proc power;
   twosamplemeans test=diff
groupmeans = 0 | 10
   stddev = 16.03
   npergroup = .
power = 0.8;
run;
```

**Power and Sample Size Calculation Using SAS Menu**

Power and sample size can also be calculated using the menu in SAS. When using the menu, the user should specify the chosen design and then fill in the required parameters needed to do the calculation for each design. The general procedure of using the menu is as follows:

1) Open SAS and Go to the enhanced editor window
2) Click the 'solutions' button on the menu.
3) In the submenu, click 'analysis'.
4) In the next submenu, click 'analyst', then a new window will pop-up.
5) In the new window, click 'statistics' button on the menu.
6) Select 'Sample size', then select the design you want to use. (Designs available in that menu include: one-sample t-test, paired t-test, two sample t-test and one-way ANOVA).
7) After you select the design another window pops-up and asks you to input the needed options and parameters. If you need to know the needed sample size for your research, you can select 'N per Group', then input number of treatments, corrected sum of square, the standard deviation and the alpha level. If the researcher wants to calculate the sample size corresponding to each power level, he/she may want to specify the range and interval of power level in the ‘Power’ row in the menu.

General data manipulation

**Transposing matrices/dataset**

Simple transposition:

```
proc transpose data=work.product
   out=work.product_transposed; run;
```

Complex transposition using by groups: change wide data format into long format.

```
proc transpose data=fishdata
```
out=fishlength(rename=(col=Measurement));
var length1-length4;
by location date; run;

**Filling missing values—Proc MI**
The MI procedure provides three methods to create imputed data sets that can be analyzed using standard procedures. The following statements are available in PROC MI:

```
PROC MI < options >;
   BY variables; EM < options >; FREQ variable; MCMC < options >;
   MONOTONE < options >; TRANSFORM transform (variables); VAR variables;
```

The EM statement uses the EM algorithm to compute the maximum likelihood estimate (MLE) of the data with missing values, assuming a multivariate normal distribution for the data. The MCMC statement uses a Markov chain Monte Carlo method to impute values for a data set with an arbitrary missing pattern, assuming a multivariate normal distribution for the data. The MONOTONE statement specifies monotone methods to impute variables for a data set with a monotone missing pattern. Available options for the MONOTONE statement include REGRESSION, REGPMM, and PROPENSITY. The TRANSFORM statement lists the variables to be transformed before the imputation process. The imputed values of these transformed variables will be reverse-transformed to the original forms before the imputation. The VAR statement lists the numeric variables to be analyzed. If you omit the VAR statement, all numeric variables not listed in other statements are used. For more information, please refer to “http://support.sas.com/rnd/app/papers/multipleimputation.pdf”.

**Delete columns & rows**
If you only want to delete variables X and Y from your data set, enter: DROP X Y;
If you only want to keep variables X and Y, enter: KEEP X Y;
If you want to delete rows where X is equal to 1, enter: IF X EQ 1 THEN DELETE;
If you want to keep all rows where X is not equal to 1, enter: IF X NE 1 THEN OUTPUT;

**More sophisticated programming tasks**

**DO Blocks and DO Loops for iterative procedure**
A DO block begins with the reserved word DO and ends with the reserved word END. The statements enclosed inside DO..END are called a block. The DO..END construct is an important device to group statements inside a DATA step. If the condition in the IF statement is true, SAS executes the statements in the DO..END block. Otherwise the statements are ignored.

```
data two; set one;
   if x < 0 then do;
      y = .;
      z = sqrt(-x) + rannor(123);
      Tx = 'Control';
   end; run;
```

**Nested IF .. THEN .. ELSE statements**

```
data two; set one;
   if score < 4 then rating='below ave';
   else if score < 6 then rating = 'average';
   else if score < 8 then rating = 'above ave.';
   else rating = 'superior'; run;
```

**Macros**
With macros you can make one small change in your program and have SAS echo that change throughout your program. Macros can also allow you to write a piece of code and use it over and over again in the same program or in different programs. You can make your programs data driven, letting SAS decide what to do. For more information regarding SAS Macro programming, please refer “http://www2.sas.com/proceedings/sugi29/243-29.pdf”

**Smoothing parameter selection**

*Smoothing with PROC MIXED*
The example describes the use of the MIXED procedure for fitting nonparametric or semi-parametric regression models.

```sas
proc mixed data=Vasca method=reml;
   model tfr_s=rp_s|rp_s/solution outpred=Smooth;
   parms (0.0015) (0.015); *parms (0.0015) (0.015)/hold=1;
   random rp_1-rp_8 /solution type=toep(1);
   where group=0;
quit; run;
```

METHOD= specifies the estimation method for the covariance parameters. The most popular methods being REML (restricted maximum likelihood) the default method, ML (maximum likelihood), MIVQUE0 (minimum variance quadratic unbiased estimation) MODEL specifies the equation for fixed effects. SOLUTION requests that a solution for the fixed-effects parameters ($\beta_0, \beta_1, \beta_2$) be produced and OUTPRED = specifies an output dataset (Smooth) containing EBLUP (pred) and related quantities (Lower, Upper – lower and upper t-type confidence limits for estimated smooth curve (EBLUP) at all study points PARMS specifies initial values for the covariance parameters and HOLD= specifies which parameters values should be held equal to the specified values. RANDOM defines the random effects constituting and SOLUTION requests that a solution for the random-effects parameters ($u_1, u_2… u_8$) be produced, TYPE= specifies the covariance structure of $G=var(u)$ and TOEP(1) specifies $\sigma_u^2 I$ structure, where I is an identity matrix.

*Automatic Smoothing parameter selection with PROC LOESS*

```sas
proc loess data=notReal; 
   model y= x1/ smooth=0.1 0.3 0.4 select=AICC1; run;
```

If you specify the SMOOTH= option in the MODEL statement, then only the values in this list are examined in performing the selection. For example, the statements select the model that minimizes the AICC1 criterion among the three models with smoothing parameter values 0.1, 0.3, and 0.4. If you do not specify the SMOOTH= option in the model statement, then by default PROC LOESS uses a golden section search method to find a local minimum of the specified criterion in the range (0,1). You can use the RANGE (lower, upper) modifier in the SELECT= option to change the interval in which the golden section search is performed. For example, the following statements request a golden section search to find a local minimizer of the GCV criterion for smoothing parameter values in the interval [0.1, 0.5]:

```sas
proc loess data=notReal; 
   model y=x1/select=GCV( range(0.1,0.5) ); run;
```

If you want to obtain a global minimum in the range of smoothing parameter values examined, you can specify the GLOBAL modifier in the SELECT= option. For example, the following statements request that a global minimizer of the AICC criterion be obtained for smoothing parameter values in the interval [0.2, 0.8]:

```sas
proc loess data=notReal; 
   model y=x1/select=AICC( global range(0.2,0.8) ); run;
```