Analyzing and Reporting Data with SAS®

The purpose of this training session is to familiarize you with ways to analyze and present oata using SAS 9.2. It is assumed you are using SAS on the Virtual Desktop. These notes build on the instructions and hints provided at the first two sessions and uses copyrighted examples developed by TexaSoft®.

START-UP TASKS:

1. Log into a lab computer using your HwakID and password.
2. Copy sample data from the CD provided by the instructor to your H:\drive or drag the folder to your desktop if your H:\ drive is not available.
3. Log in to the Virtual Desktop using your HawkID and password.
4. Click on SAS 9.2 icon to start up a SAS session.

Using PROC MEANS

PROC MEANS produces descriptive statistics (means, standard deviation, minimum, maximum, etc.) for numeric variables in a set of data. PROC MEANS can be used for

- Describing continuous data where the average has meaning
- Describing the means across groups
- Searching for possible outliers or incorrectly coded values
- Performing a single sample t-test

The **syntax** of the PROC MEANS statement is:

PROC MEANS <options>; <statements>;

Default statistical options are N, MEAN, STD, MIN, MAX. Other commonly used **options** available in PROC MEANS include DATA=, NOPRINT, MAXDEC=*n*. Commonly used **statements** with PROC MEANS include

- **BY** variable list -- Statistics are reported for groups in separate tables
- **CLASS** variable list – Statistics reported by groups in a single table
- **VAR** variable list – specifies which numeric variables to use
- **OUTPUT OUT** = data set name – statistics will be output to a SAS data file
- **FREQ** variable - specifies a variable that represents a count of observations

A few quick examples of PROC MEANS:

```
* Simplest invocation – on all numeric variables *;
PROC MEANS;

*Specified statistics and variables *;
PROC MEANS N MEAN STD;
   VAR SODIUM CARBO;
```

```
* Subgroup descriptive statistics using by statement*;
PROC SORT; BY SEX;
PROC MEANS; BY SEX;
   VAR FAT PROTEIN SODIUM;

* Subgroup descriptive statistics using class statement*;
PROC MEANS;
   CLASS SEX;
VAR FAT PROTEIN SODIUM;
```

Example 1: A simple use of PROC MEANS

This example calculates the means of several specified variables, limiting the output to two decimal places (uses SAS program file **procmeans1**).

```
***************************************************************

* Data on weight, height, and age of a random sample of 12    *
* nutritionally deficient children                            *
***************************************************************;
DATA CHILDREN;
INPUT WEIGHT HEIGHT AGE;
DATALINES;
64 57 8
71 59 10
53 49 6
67 62 11
55 51 8
58 50 8
77 55 10
57 48 9
56 42 10
51 42 6
76 61 12
68 57 9
;
ODS RTF;
proc means;
Title 'Example 1a - PROC MEANS, simplest use';
run;

proc means maxdec=2;var WEIGHT HEIGHT;
Title 'Example 1b - PROC MEANS, limit decimals, specify variables';
run;

proc means maxdec=2 n mean stderr median;var WEIGHT HEIGHT;
Title 'Example 1c - specify statistics to report';
run;
ODS RTF CLOSE;
```

Output for Example 1:

```
                  Example 1a - PROC MEANS, simplest use

Variable       N        Mean      Std Dev      Minimum       Maximum
WEIGHT        12  62.7500000    8.9861004   51.0000000    77.0000000
HEIGHT        12  52.7500000    6.8240884   42.0000000    62.0000000
AGE           12   8.9166667    1.8319554    6.0000000    12.0000000

        Example 1b - PROC MEANS, limit decimals, specify variables

        Variable       N    Mean   Std Dev    Minimum    Maximum
        WEIGHT        12   62.75      8.99      51.00      77.00
        HEIGHT        12   52.75      6.82      42.00      62.00

          Example 1c – PROC MEANS, specify statistics to report

            Variable       N    Mean   Std Error   Median
            WEIGHT        12   62.75        2.59    61.00
            HEIGHT        12   52.75        1.97    53.00
```

Example 2: Using PROC MEANS with BY and CLASS statements

This example uses PROC MEANS to calculate means for an entire data set or by a grouping variable (uses SAS program file **procmeans2**).

```
***********************************
*    Example 2 for _Proc Means    *
***********************************;
DATA FERTILIZER;
INPUT FEEDTYPE WEIGHTGAIN;
DATALINES;
1 46.20
1 55.60
1 53.30
1 44.80
1 55.40
1 56.00
1 48.90
2 51.30
2 52.40
```

```
2 54.60
2 52.20
2 64.30
2 55.00
;
ODS RTF;
PROC SORT DATA=FERTILIZER;BY FEEDTYPE;
PROC MEANS; VAR WEIGHTGAIN; BY FEEDTYPE;
TITLE 'Summary statistics by group';
RUN;
PROC MEANS; VAR WEIGHTGAIN; CLASS FEEDTYPE;
TITLE 'Summary statistics USING CLASS';
RUN;
ODS RTF CLOSE;
;
```

Output for this SAS code is:

Summary Statistics by Group

FEEDTYPE=1

Analysis Variable : WEIGHTGAIN

N	Mean	Std Dev	Minimum	Maximum
7	51.4571429	4.7475808	44.8000000	56.0000000

FEEDTYPE=2

Analysis Variable : WEIGHTGAIN

N	Mean	Std Dev	Minimum	Maximum
6	54.9666667	4.7944412	51.3000000	64.3000000

In this first example the BY statement working with the PROC SORT creates two tables -- one for each value of the BY variable. In this next example, the CLASS statement produces a single table broken down by group (FEEDTYPE.)

Summary statistics USING CLASS

Analysis Variable : WEIGHTGAIN

FEEDTYPE	N Obs	N	Mean	Std Dev	Minimum	Maximum
1	7	7	51.4571429	4.7475808	44.8000000	56.0000000

```
                        Analysis Variable : WEIGHTGAIN
               N
FEEDTYPE     Obs    N        Mean       Std Dev      Minimum       Maximum
       2       6    6   54.9666667    4.7944412   51.3000000    64.3000000
```

Hands-on Exercise:

1. Modify the above program to output the following statistics:
   N MEAN MEDIAN MIN MAX
2. Use MAXDEC=2 to limit the number of decimals in output.

EXAMPLE 3: Using PROC MEANS to find OUTLIERS

PROC MEANS is a quick way to find large or small values in your data set that may be considered outliers (see PROC UNIVARIATE also.) This example shows the results of using PROC means where the MINIMUM and MAXIMUM identify unusual values in the data set. (uses SAS program file **procmeans3**).

```
**************************************************
* USING PROC MEANS TO FIND OUTLIERS              *
*************************************************;
DATA WEIGHT;
INPUT TREATMENT LOSS @@;
DATALINES;
2 1.0 1 3.0 1 -1.0 1 1.5 1 0.5 1 3.5 1 -99
2 4.5 3 6.0 2 3.5 2 7.5 2 7.0 2 6.0 2 5.5
1 1.5 3 -2.5 3 -0.5 3 1.0 3 .5 3 78 1 .6 2 3 2 4 3 9 1 7 2 2
;
ODS RTF;
PROC MEANS; VAR LOSS;
TITLE 'Find largest and smallest values';
RUN;
ODS RTF CLOSE;
```

Notice in this output, PROC means indicates there is a small value of -99 (could be a missing value code) and a large value of 78 (could be a miscoded number). This is a quick way to find outliers in your data set.

```
                 Find Largest and Smallest Values

                    Analysis Variable : LOSS
 N        Mean        Std Dev        Minimum        Maximum
26   2.0423077     25.4650062    -99.0000000     78.0000000
```

EXAMPLE 4: Using PROC MEANS to perform a single sample t-test (or Paired t-test)

To compare two paired groups (such as in a before-after situation) where both observations are taken from the same or matched subjects, you can perform a paired t-test using PROC MEANS. To do this convert the paired data into a difference variable and perform a single sample t-test. For example, suppose your data contained the variables WBEFORE and WAFTER, (before and after weight on a diet), for 8 subjects. To perform a paired t-test using PROC MEANS, follow these steps:

1. Read in your data.
2. Calculate the difference between the two observations (WLOSS is the amount of weight lost), and
3. Report the mean loss, t-statistic and p-value using PROC MEANS.

The hypotheses for this test are:

Ho: µLoss = 0 (The average weight loss was 0)
Ha: µLoss ≠ 0 (The weight loss was different than 0)

For example, the following code performs a paired t-test for weight loss data:

(uses SAS program file **procmeans4**)

```
********************************************************
* Paired t-test/Single sample t-test                   *
********************************************************
;
DATA WEIGHT;
INPUT WBEFORE WAFTER;
* Calculate WLOSS in the DATA step *;
WLOSS=WAFTER-WBEFORE;
DATALINES;
200 190
175 154
188 176
198 193
197 198
310 240
245 204
202 178
;
ODS RTF;
PROC MEANS N MEAN T PRT; VAR WLOSS;
TITLE 'Paired t-test example using PROC MEANS';
RUN;
ODS RTF CLOSE;
```

Notice that the actual test is performed on the new variable called WLOSS, and that is why it is the only variable requested in the PROC MEANS statement. This is essentially a one-sample t-test. The statistics of interest are the mean of WLOSS, the t-statistic associated with the null hypothesis for WLOSS and the p-value. The SAS output is as follows:

Paired t-test example using PROC MEANS

```
          Analysis Variable : WLOSS
 N              Mean    t Value     Pr > |t|
 8       -22.7500000      -2.79       0.0270
```

The mean of the variable WLOSS is –22.75. The t-statistic associated with the null hypothesis is –2.79, and the p-value for this paired t-test is p = 0.027, which provides evidence to reject the null hypothesis.

EXAMPLE 5: Using PROC MEANS to output additional statistics

Suppose you have a data set and you want to add a column containing a z-statistic based on the mean and standard deviation of a variable. Here is one way to do that. The following data set contains weights of 12 children. You want to add a column of the difference of the scores from the mean based on a the information in the WEIGHT variable. For good measure also calculate the z-score (uses SAS program file **procmeans5**).

```
*******************************************************
* PROC MEANS 5 Example                                       *
* (C) Alan Elliott                                                     *
*******************************************************;
DATA WT;
INPUT WEIGHT;
DATALINES;
64
71
53
67
55
58
77
57
56
51
76
68
;
PROC MEANS NOPRINT DATA=WT;VAR WEIGHT;OUTPUT OUT=WTMEANS MEAN=WTMEAN
STDDEV=WTSD;
RUN;
DATA WTDIFF;SET WT;
IF _N_=1 THEN SET WTMEANS;
DIFF=WEIGHT-WTMEAN;
Z=DIFF/WTSD; * CREATES STANDARDIZED SCORE (Z-SCORE);
RUN;
ODS RTF;
PROC PRINT DATA= WTDIFF;VAR WEIGHT DIFF Z;
```

```
RUN;
ODS RTF CLOSE;

*Bonus code -- get a zscore using PROC STANDARD;

PROC STANDARD DATA=WT
    MEAN=0 STD=1 OUT=ZSCORES;
    VAR WEIGHT;
RUN;
PROC PRINT DATA=ZSCORES;
RUN;
```

The statement OUTPUT OUT=WTMEANS MEAN=WTMEAN STDDEV=WTSD; creates a SAS data file containing a single record with variables WTMEAN and WTSD (and some other system variables.) You can then use that information to calculate the desired values, as is done in the code:

```
DATA WTDIFF;SET WT;
IF _N_=1 THEN SET WTMEANS;
DIFF=WEIGHT-WTMEAN;
Z=DIFF/WTSD; * CREATES STANDARDIZED SCORE (Z-SCORE);
RUN;
```

The first SET statement (SET WT) reads in the entire WT data set. The statement

```
IF _N_=1 THEN SET WTMEANS;
```

reads in the first (and only) record from the WTMEANS data set and merges theWTDIFF and WTSD (and a couple of other system variables) into the new WTDIFF dataset, allowing you to do the calculations to come up with the DIFF and Z values.

The resulting data set contains the following information

Obs	WEIGHT	DIFF	Z
1	64	1.25	0.13910
2	71	8.25	0.91808
3	53	-9.75	-1.08501
4	67	4.25	0.47295
5	55	-7.75	-0.86244
6	58	-4.75	-0.52859
7	77	14.25	1.58578
8	57	-5.75	-0.63988
9	56	-6.75	-0.75116
10	51	-11.75	-1.30757
11	76	13.25	1.47450
12	68	5.25	0.58424

NOTE: You could also get standardized values using PROC STANDARD.

```
PROC STANDARD DATA=WT
MEAN=0 STD=1 OUT=ZSCORES;
VAR WEIGHT;
RUN;
PROC PRINT DATA=ZSCORES;
RUN;
```

Using PROC UNIVARIATE

If the PROC MEANS procedure does not produce the statistic you need for a data set then PROC UNIVARIATE may be your choice. Although it is similar to PROC MEANS, its strength is in calculating a wider variety of statistics, specifically useful in examining the distribution of a variable.

Use PROC UNIVARIATE to examine the distribution of your data, including an assessment of normality and discovery of outliers.

The syntax of the PROC UNIVARIATE statement is:

PROC UNIVARIATE <options>; <statements>;

Commonly used options for PROC UNIVARIATE include:

DATA= - Specifies data set to use
NORMAL - Produces a test of normality
FREQ – Produces a frequency table
PLOT – Produces stem-and-leaf plot

Commonly used statements used with PROC UNIVARIATE include:

BY variable list;
VAR variable list;
OUTPUT OUT = datasetname;

The BY group specification causes UNIVARIATE to calculate statistics separately for groups of observations (i.e., treatment means). The OUTPUT OUT= statement allows you to output the means to a new data set. The following SAS program (uses SAS program file **procuni1**) produces a large number of statistics on the variable AGE:

```
DATA EXAMPLE;
INPUT TREATMENT LOSS @@;
DATALINES;
;
PROC UNIVARIATE NORMAL PLOT DATA=EXAMPLE; VAR AGE;
HISTOGRAM AGE/NORMAL (COLOR=RED W=5);
TITLE 'PROC UNIVARIATE EXAMPLE';
FOOTNOTE 'Evaluate distribution of variables';
RUN;
```

The output from this program follows. The first table gives standardized descriptive statistics (Moments). These statistics allow you to gain an idea of the distribution of data within the variable AGE.

Moments			
N	50	**Sum Weights**	50
Mean	10.46	**Sum Observations**	523
Std Deviation	2.42613323	**Variance**	5.88612245
Skewness	-0.5119219	**Kurtosis**	-0.2610615
Uncorrected SS	5759	**Corrected SS**	288.42
Coeff Variation	23.1943903	**Std Error Mean**	0.34310705

The next table provides basic measures of central tendency and spread.

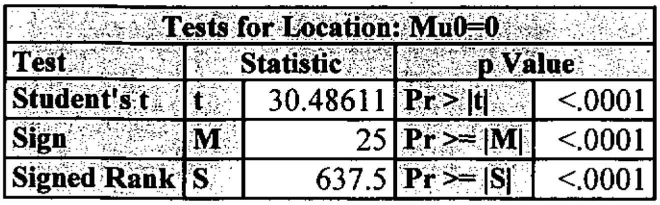
Basic Statistical Measures			
Location		Variability	
Mean	10.46000	**Std Deviation**	2.42613
Median	11.00000	**Variance**	5.88612
Mode	12.00000	**Range**	11.00000
		Interquartile Range	3.00000

The table "Tests for location" provides a test for the null hypothesis that the mean is zero. This can be used for a paired value (paired t-test using Student's t_ to test.

H_o: m = 0 (The mean is 0)

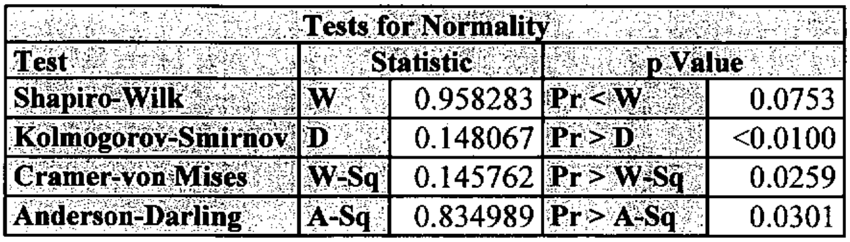
H_a: m ≠ 0 (The mean differs from 0)

The Sign test and Signed rank tests are nonparametric tests.



Tests for Location: Mu0=0				
Test	Statistic		p Value	
Student's t	t	30.48611	Pr > |t|	<.0001
Sign	M	25	Pr >= |M|	<.0001
Signed Rank	S	637.5	Pr >= |S|	<.0001

The test for normality are one way of assessing whether the distribution of the data appears normally distributed. Four tests for normality are provided:



Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.958283	Pr < W	0.0753
Kolmogorov-Smirnov	D	0.148067	Pr > D	<0.0100
Cramer-von Mises	W-Sq	0.145762	Pr > W-Sq	0.0259
Anderson-Darling	A-Sq	0.834989	Pr > A-Sq	0.0301

Notice that in this case these test differ in outcome (assuming a criteria of 0.05 is strictly followed) with the Shapiro-Wilk test providing evidence that the data are normally distributed (p=0.075) while the others reject this hypothesis.

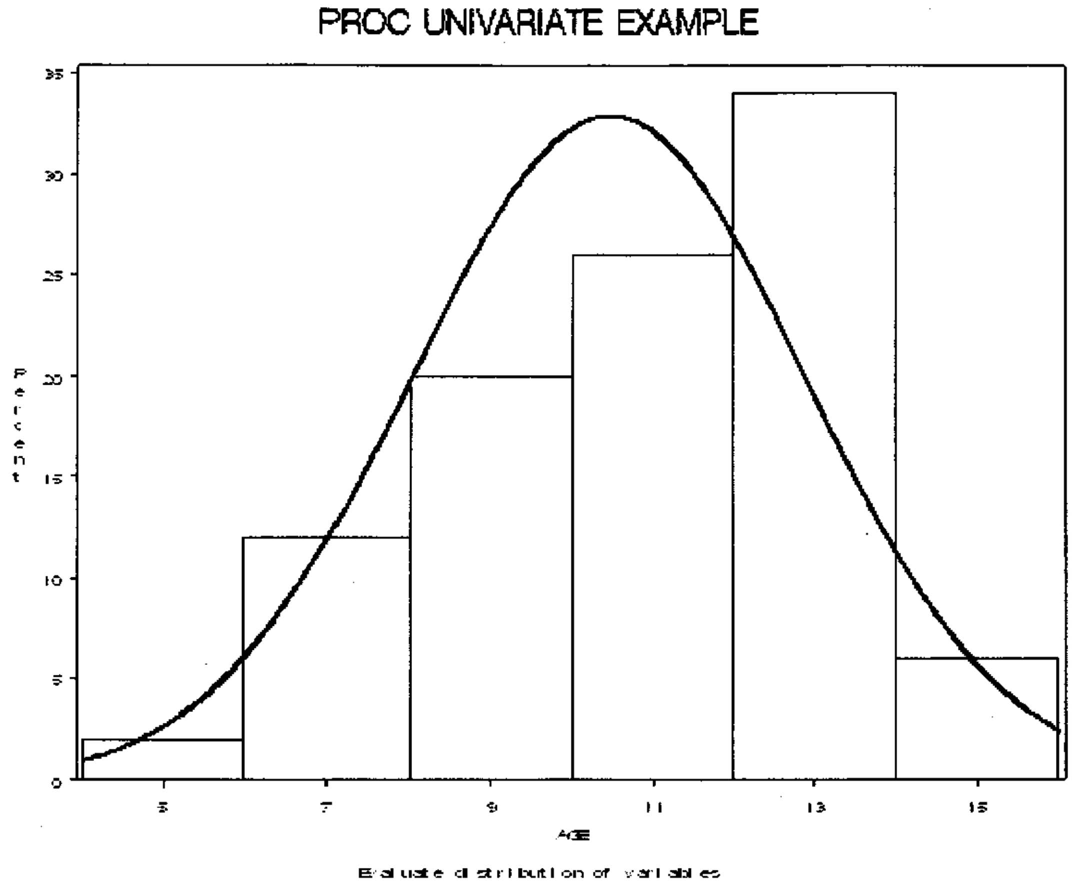
The inclusion of the NORMAL and PLOT statement in

```
PROC UNIVARIATE NORMAL PLOT DATA=EXAMPLE; VAR AGE;
```

provides the test for normality plus a box and whiskers plot and a stem and leaf diagram.

Additional output that is useful is visually assessing normality may be created by including one the HISTOGRAM statement as shown below:

```
PROC UNIVARIATE NORMAL PLOT DATA=EXAMPLE; VAR AGE;
HISTOGRAM AGE/NORMAL (COLOR=RED W=5);
```



The superimposed normal plot on the histogram allows you to not only see if the data are approximately normally distributed, it also shows where it may not be fitting normality. In this case, it appears that the plot has more than expected values at the upper end of the range.

Evaluating more than one category of a variable

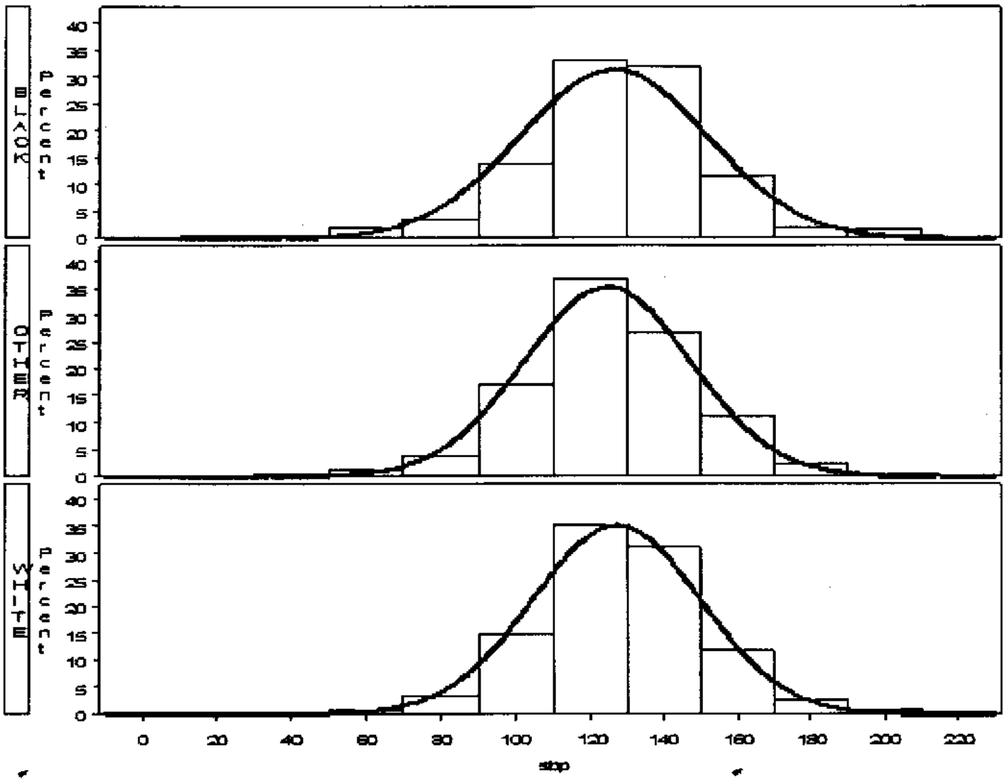
Suppose you have several groups that you are comparing and you want to examine the distribution of the variable by group. The following example provides examples of how you could create histograms by RACE_CATEGORY using PROC UNIVARIATE (uses SAS data file **procuni2**).

```
PROC UNIVARIATE DATA=SASDATA2.SBPDATA NOPRINT;
  CLASS RACE_CATEGORY;
  VAR SBP;
  HISTOGRAM /NORMAL (COLOR=RED W=5) NROWS=3;
RUN;
```

In this example data is from a trauma data set (SBPDATA extracted from the National Trauma Data set, 2004). The new statements used in this example include:

- NOPRINT – since we're only interested in producing the graph, this option suppress other output
- CLASS RACE_CATEGORY -- This statement indicates that the data is to be examined for each category (classification) of the RACE_CATEGORY variable.
- ROWS=3 -- Since we know that there are three categories (BLACK, WHITE and OTHER), we add the option "NROWS=3" to the HISTOGRAM statement to indicate how many graphs to put on a singe page.

The following plot is created:



Notice that the three histograms are for the three values of RACE_CATEGORY which are BLACK,""OTHER," and "WHITE." This graph is helpful in comparing the distribution of data in two or more groups. In this case, there is visual agreement that SBP is similarly distributed for all races.

Graph by two factors

Suppose you have two grouping variables and you want to produce a series of histograms to compare distributions. The following program (uses SAS program file **procuni3**) produces a series of histograms by GENDER and WOUND type. Since this is a more detailed program the parts are annotated and described below:

```
PROC FORMAT;
VALUE FMTWOUND 0="NONPENETRATING"
               1="PENETRATING";
RUN;
TITLE 'HISTOGRAMS of SBP by GENDER and WOUND TYPE';
PROC UNIVARIATE DATA=SASDATA2.SBPDATA NOPRINT;
  CLASS WOUND GENDER;
  VAR SBP;
  HISTOGRAM / NROWS=2 NCOLS=2 CFILL=BLUE PFILL=M2N45;
  INSET N='N:' (4.0) MIN='MIN:' (4.1) MAX='MAX:' (4.1) / NOFRAME POSITION=NE
HEIGHT=2;
  FORMAT WOUND FMTWOUND.;
RUN;
```

PROC FORMAT – this procedure creates a format for the WOUND variable to describe the coded 0,1 variables. Using this format allows you to display the groups in the graph by clearer category names (PENETRATE and NONPENETRATE) than by the cryptic 0 and 1. (See Chapter 3 for more information on PROC FORMAT.)

TITLE statement – this places a title at the top of the graph. If you use other title statements such as TITLE2, the subsequent titles will be smaller by default than the first title (unless you change that in code.) (See chapter 3 for more information on titles.)

CLASS statement – In this example there are two grouping variables indicated in the CLASS statement.

```
CLASS WOUND GENDER;
```

HISTOGRAM STATEMENT -- The options within the HISTOGRAM statement define how the graph will appear. The columns and rows: The statements

```
NROWS=2 NCOLS=2
```

produce 2 histograms per row (for WOUND – first item in the CLASS statement) and 2 histograms for per COL (for GENDER or 2nd item in the CLASS statement)

The histogram bar colors are specified by the CFILL (color fill) statement:

```
CFILL=BLUE
```

In this case, the bars will be blue. Some of the colors available in SAS (there are thousands to choose from) include

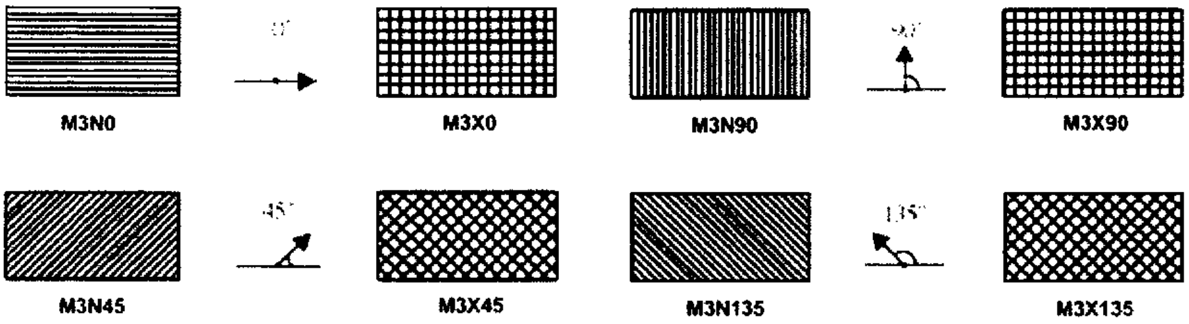
```
BLACK     WHITE     RED       GREEN     BLUE      PURPLE
VIOLET    ORANGE    YELLOW    PINK      CYAN      MAGENTA
BROWN     GOLD      LIME      GRAY      LILAC     MAROON
SALMON    TAN       ROSE      CREAM
```

The default color is black.

The pattern for the bars is specified by the PFILL (Pattern fill) statement

```
PFILL=M3N45
```

You can select from a number of available patterns. The default pattern is solid. Here are some of the other patterns you can select:



INSET option – this defines an inset or key to the graph. This example illustrates several of the options:

```
INSET N='N:' (4.0) MIN='MIN:' (4.1) MAX='MAX:' (4.1)
               / NOFRAME POSITION=NE HEIGHT=2;
```

The statement

```
N='N:' (4.0) MIN='MIN:' (4.1) MAX='MAX:' (4.1)
```

defines which statistics will be included in the inset. In this case N (the sample size) will be designated with "N:" and will be displayed using the SAS output format 4.0. The MIN and MAX are similarly defined.

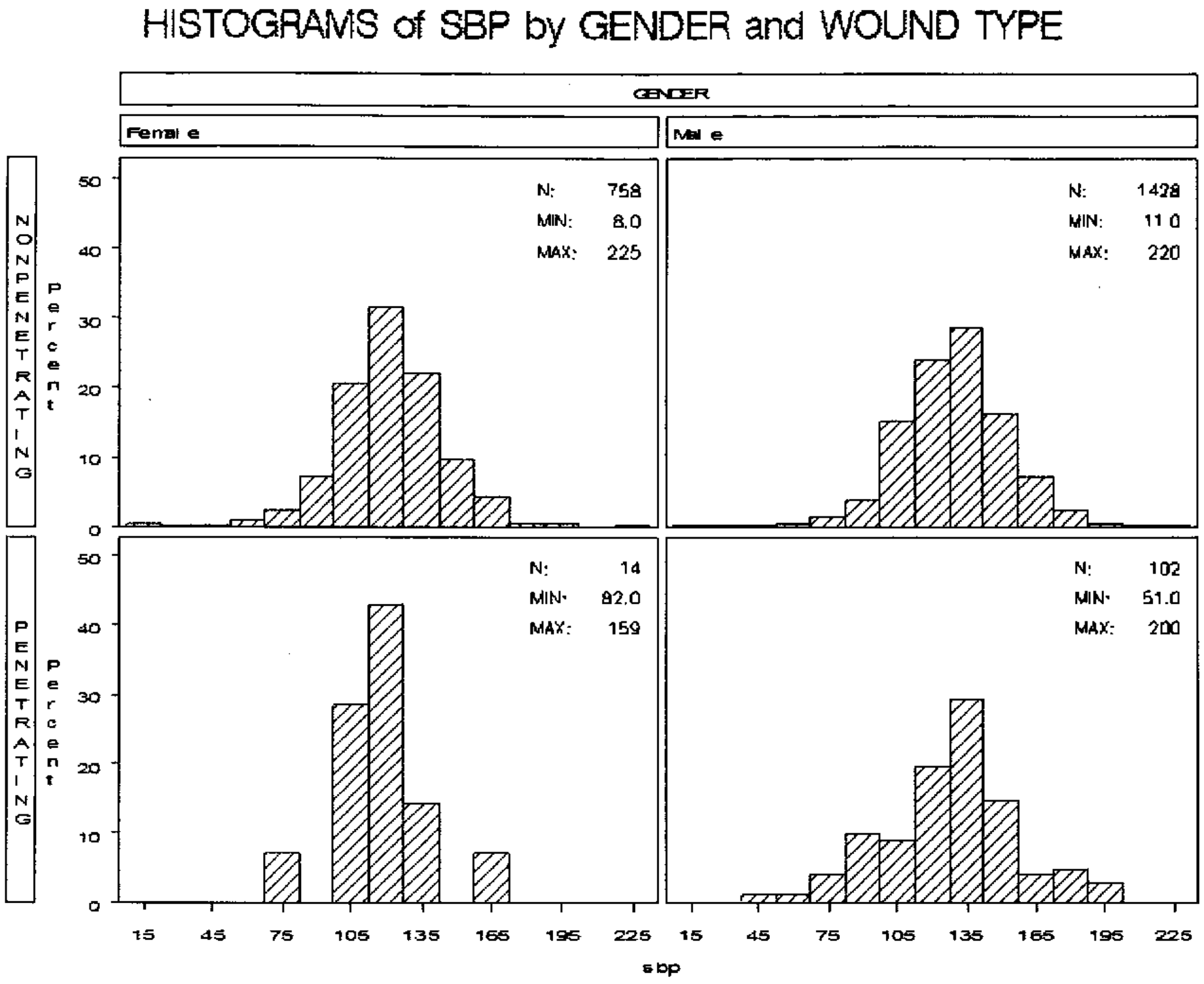
The remaining options

```
/ NOFRAME POSITION=NE HEIGHT=2;
```

specify that there be

- no frame around the inset
- that its position will be in the NE = North-East corner of the graph
- and that the height of the characters will be set at 2 units.

When this SAS code is run, it produces the following graphs:



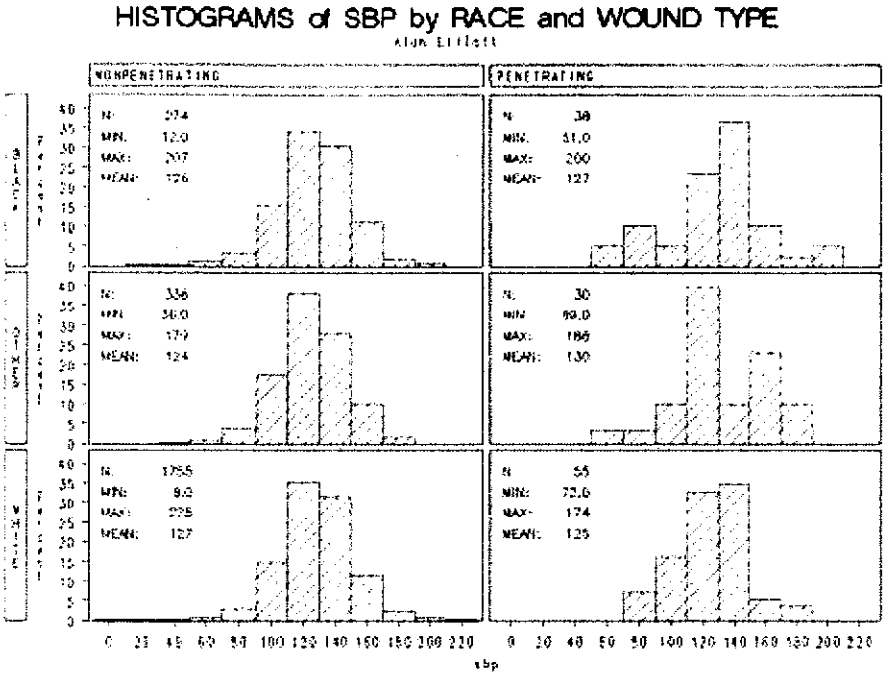
Exercise: Experiment with the colors, patterns and inset to see how they affect the graph.

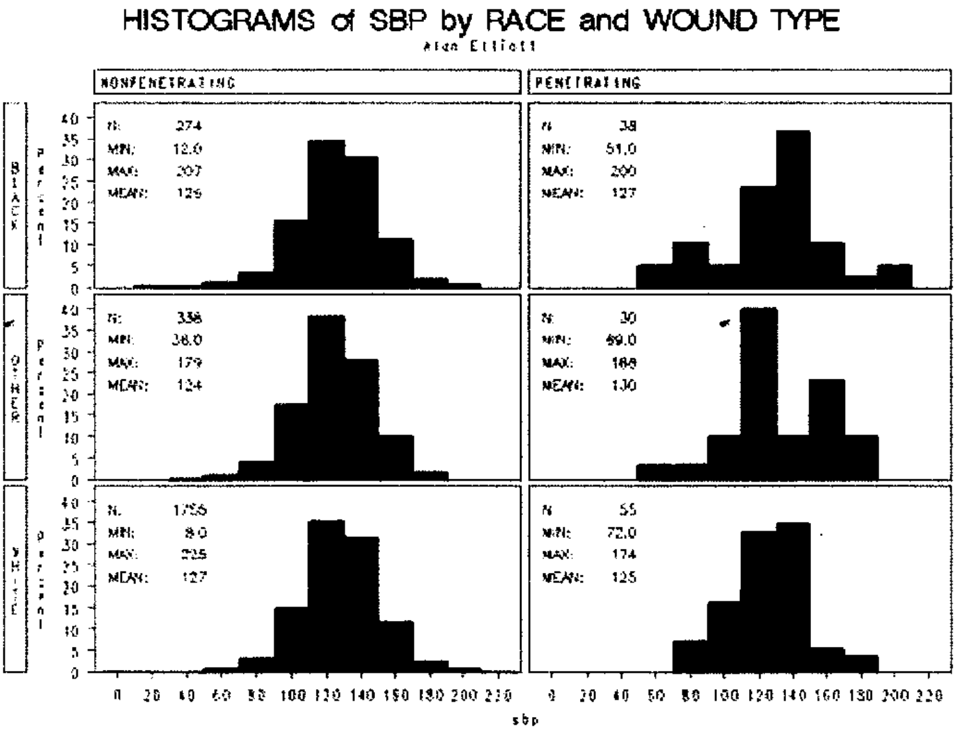
1. Make the histogram color Green
2. Add the option MEAN='MEAN:' (4.1) to the inset option.
3. Add the NORMAL(COLOR=BROWN W=3)statement to superimpose a normal plot
4. How does this change the plot?

Exercise: Using the SBPDATA create the following histograms:

1. Create a matrix of histograms with RACE_CATEGORY (3 categories) using the pattern M3XO and CFILL=RED.
2. Place the key on the upper left corner (NW).
3. Add MEAN='MEAN:' (4.1) to the list of statistics reported.
4. Put your name in a TITLE2 statement.
5. Redo the plot using a solid blue bars.
6. Capture the output using ODS PDF and print the results.

The resulting graphs should look like this:





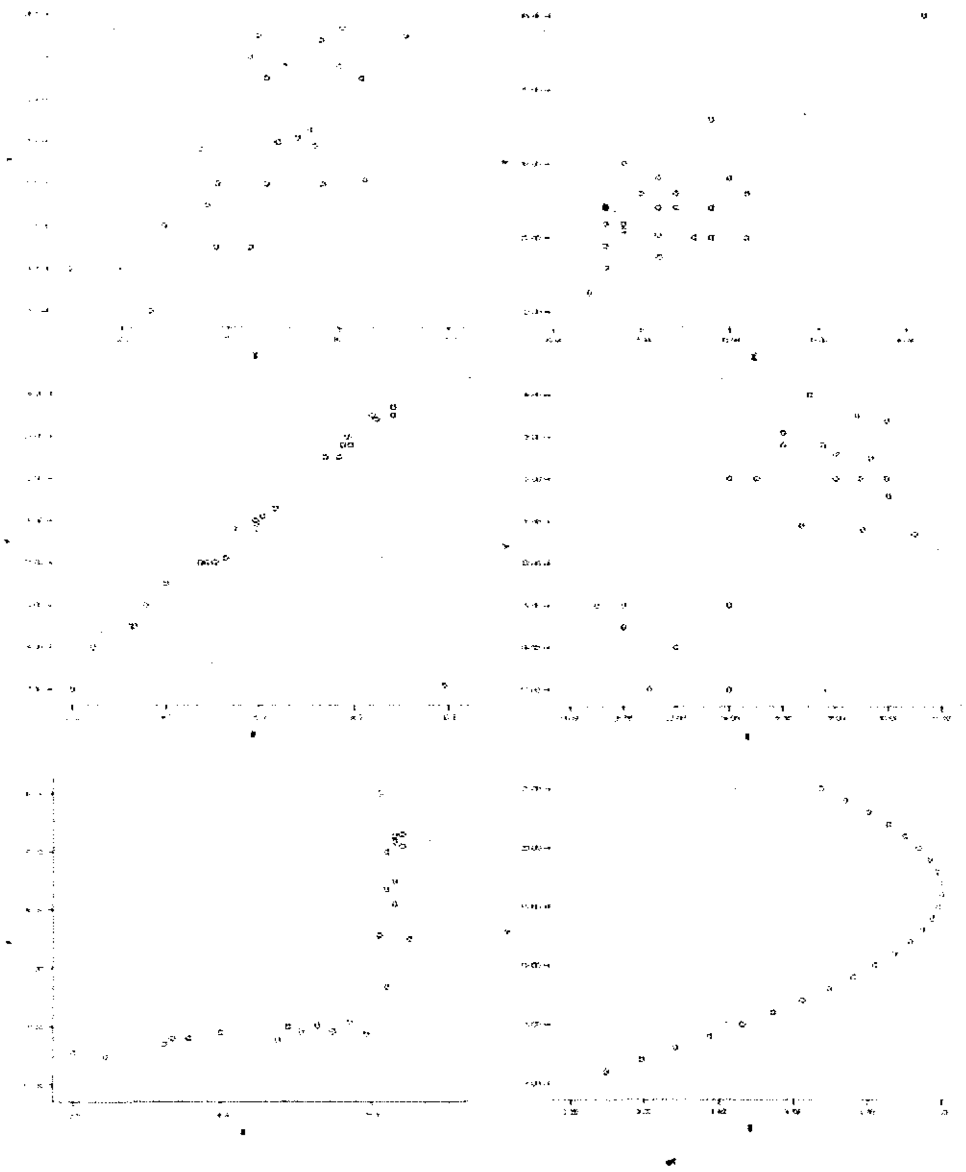
Correlation Analysis using PROC CORR

The correlation coefficient allows researchers to determine if there is a possible linear relationship between two variables measured on the same subject (or entity). When these two variables are of a continuous nature (they are measurements such as weight, height, length, *etc.*) the measure of association most often used is Pearson's correlation coefficient.

This association may be expressed as a number (the correlation coefficient) that ranges from −1 to +1. The population correlation is usually expressed as the Greek letter *rho* (⍰) and the sample statistic (correlation coefficient) is r.

The correlation measures how well a straight line fits through a scatter of points when plotted on an x – y axis. If the correlation is positive, it means that when one variable increases, the other tends to increase. If the correlation is negative, it means that when one variable increases, the other tends to decrease. When a correlation coefficient is close to +1 (or −1), it means that there is a strong correlation – the points are scattered along a straight line. For example, a correlation $r = 0.7$ may be considered strong. However, the closer a correlation coefficient gets to 0, the weaker the relationship, where the cloud (scatter) of points is not close to a straight line. For example, a correlation $r = 0.1$ might be considered weak. For scientific purposes, a t-test is utilized to determine if the correlation coefficient is "strong" or "significant" or not. This will be discussed later.

Assumptions: Before using the Pearson correlation coefficient as a measure of association, you should be aware of its assumptions and limitations. As mentioned earlier, this correlation coefficient measures a *linear* relationship. That is, the relationship between the two variables measures how close the two measurements form a straight line when plotted on an x-y chart. Therefore, it is important that data be graphed before the correlation is interpreted. For example, it is possible that data, when plotted, may show a curved relationship instead of a straight line. When this is the case, a Pearson correlation may not be the best measure of association. There are other conditions when a correlation coefficient may appear important, but when considered in light of a graph, is not a good measure of relationship. In the following graphs, **all of them have a correlation coefficient of about 0.72,** yet most do not fit the assumption of a linear relationship. To avoid misinterpreting a correlation, always accompany the calculation with a graph.



Another assumption of correlation is that the both of the variables (the measurements) be of continuous data measured on an interval/ratio scale. Data that are not continuous, such as categorical (i.e. hair color) or binomial (i.e., gender) data would not be acceptable. Also, each variable should be approximately normally distributed.

The SAS procedure most often used to calculate correlations is PROC CORR. The syntax for this procedure is:

```
    PROC CORR <options>; <statements>;
```

The most commonly used option is

```
    DATA=datsetname;
```

The most commonly used information statements are:

```
        VAR variablelist;
        BY varlist
```

As an example, to find the correlations between variables in the SOMEDATA data set use the following program (uses SAS program file **proccorr1** in addition to the SAS data file **SOMEDATA.SAS7BDAT**.)

```
*    ASSUMES YOU HAVE A SAS LIBRARY NAMED MYDATA
*    THAT INCLUDES THE FILE SOMEDATA.SAS7BDAT;
ods rtf;
PROC CORR data=mydata.somedata;
      VAR AGE TIME1-TIME2;
TITLE 'Example correlation calculations using PROC CORR';
run;
ods rtf close;
```

The (partial) output from this program is:

"Pearson Correlations" **Pearson Correlation Coefficients, N = 50** **Prob > |r| under H0: Rho=0**			
	AGE	**TIME1**	**TIME2**
AGE Age on Jan 1, 2000	1.00000	0.50088 0.0002	0.38082 0.0064
TIME1 Baseline	0.50088 0.0002	1.00000	0.76396 <.0001
TIME2 6 Months	0.38082 0.0064	0.76396 <.0001	1.00000

The output includes descriptive statistics on each variable and a table of Pearson Correlation Coefficients (*r*). For example, the correlation between AGE and TIME1 is 0.50088, or r=0.50088. The number under each correlation is a p-value. It tests to see if *r* is statistically significant. In statistical terminology, this is a test of the following hypotheses

H_0: *rho* = 0 (the null hypothesis)

H_a: *rho* <> 0 (the alternative hypothesis)

If the p-value for the test is small (usually less than 0.05) then the conclusion is that *rho* is ***not*** 0, thus the relationship is *statistically* significant. A research will then have to make a professional judgment to determine if the association is significant in terms of the experiment performed.

Care must be taken when interpreting a statistically significant correlation. If your sample size is small or not representative of the population from which you sampled, you may not be able to generalize the correlation to your intended population. Also, a **cause and effect relationship cannot be inferred** except under special conditions when you have designed the study specifically to detect those phenomena.

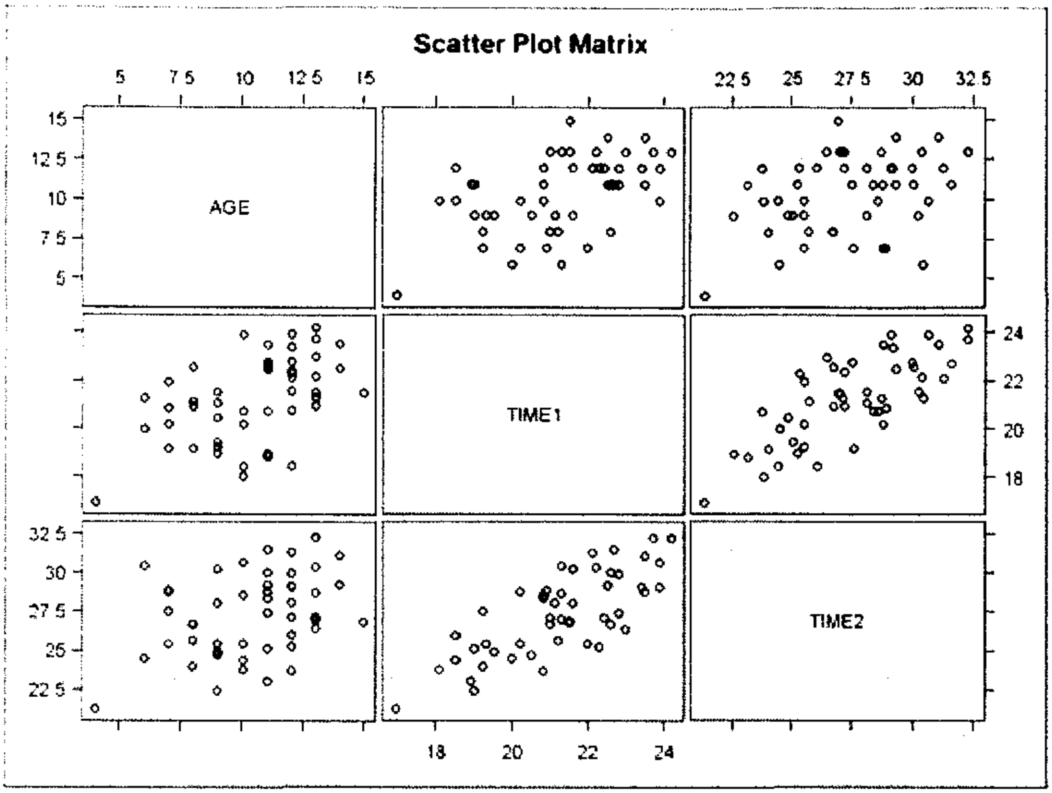
Note – to have the program output both PEARSON and SPEARMAN (non-parametric) correlations, use the statement:

```
PROC CORR data=mydata.somedata PEARSON SPEARMAN;
```

To observe a scatter plot for each correlation, use this slight variation on the program (uses SAS data file **proccorr2**). Notice the addition of the **ODS GRAPHICS** statements and **PLOTS=MATRIX**.

```
ODS RTF;
ODS GRAPHICS ON;
PROC CORR DATA=MYDATA.SOMEDATA PLOTS=MATRIX;
      VAR AGE TIME1-TIME2;
TITLE 'Example correlation calculations using PROC CORR';
RUN;
ODS RTF CLOSE;
ODS GRAPHIC OFF;
```

This produces the following matrix of scatter plots:



Note that in this plot the upper and lower half are identical – the plot is symmetric, so you really only have to look at half of it.

Creating One-Way Frequency Tables with PROC FREQ

Data that are collected as counts require a specific kind of data analysis. It doesn't make sense to calculate means and standard deviations on categorical data. Instead, categorical data is analyzed by creating frequency and cross tabulation tables. The primary procedure within SAS for this kind of analysis is PROC FREQ.

This tutorial covers the creation and analysis of a single variable frequency table using the PROC FREQ procedure.

The syntax for PROC FREQ is:

```
PROC FREQ <options>; TABLES specification; <statements>;
```

Commonly used options used in PROC FREQ is:

```
      DATA =        (Specify which data set to use)
      ORDER=FREQ    (Output data in frequency order)
```

A commonly used statement used with PROC FREQ is:

```
BY varlist  (Specify BY list to create subsetted analyses)
```

The TABLES statement is used to request which tables will be produced. For example, to obtain counts of the number of subjects in each GROUP categories, use the code:

```
PROC FREQ; TABLES GROUP;
```

To produce a chi-square test for goodness of fit, use code such as

```
PROC FREQ;
  TABLES COLOR / CHISQ NOCUM TESTP=(0.5625 0.1875 0.1875 0.0625);
```

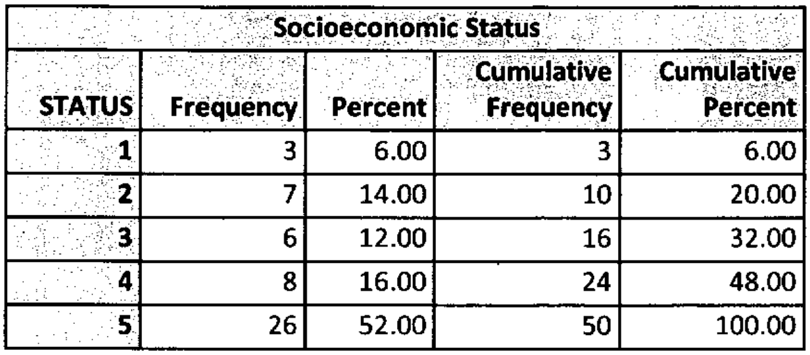
(See details about these options later in the tutorial.)

Creating a One-Way Frequency Table

When only one variable is used in the TABLES statement, PROC FREQ produces a frequency table. For example, using the data from the SOMEDATA SAS data set, the following code produces a frequency table using data in the STATUS variable (uses SAS program file **procfreq1**).

```
* ASSUMES YOU HAVE A SAS LIBRARY NAMED MYDATA;
ODS RTF;
PROC FREQ DATA=MYDATA.SOMEDATA; TABLES STATUS;
TITLE 'Simple Example of PROC FREQ';
RUN;
PROC FREQ DATA=MYDATA.SOMEDATA ORDER=FREQ; TABLES STATUS;
TITLE 'Simple Example of PROC FREQ';
RUN;
ODS RTF CLOSE;
```

The output for this job is:



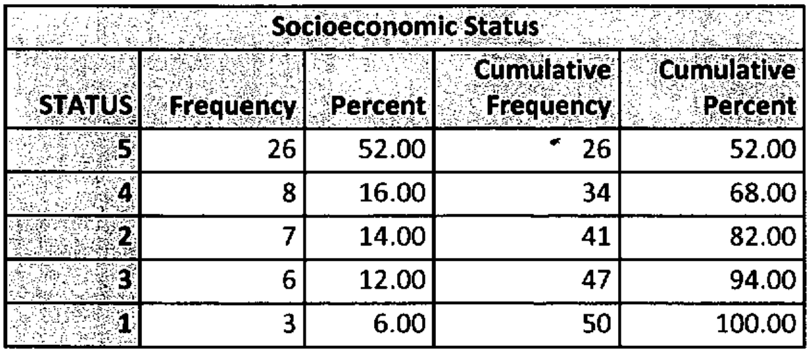
Socioeconomic Status				
STATUS	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	3	6.00	3	6.00
2	7	14.00	10	20.00
3	6	12.00	16	32.00
4	8	16.00	24	48.00
5	26	52.00	50	100.00

The frequency gives the count of the number of times the STATUS variable took on the value in the STATUS column. The percent column is the percent of total (50). The Cumulative Frequency and Percent columns report an increasing count or percent for each value of STATUS. Use this type of analysis to discover the distribution of the categories in your data set. For example, in this data, over half of the subjects fall into the STATUS=5 category. If you'd hoped for a representative sample in each category, this shows you that that criteria was not met.

Exercise: Using the Order=Freq orders the table by frequency. Change the PROC FREQ line to read

```
PROC FREQ Order=Freq; TABLES STATUS;
```

And rerun the program to get the sorted by frequency output. This helps you identify which categories have the most and fewest counts.



Socioeconomic Status				
STATUS	Frequency	Percent	Cumulative Frequency	Cumulative Percent
5	26	52.00	26	52.00
4	8	16.00	34	68.00
2	7	14.00	41	82.00
3	6	12.00	47	94.00
1	3	6.00	50	100.00

Suppose your data were already summarized into counts .In this case you can use the WEIGHT statement to read in your data ((uses SAS program file **procfreq2**). For example:

```
DATA CDS;
     INPUT @1 CATEGORY $9. @10 NUMBER 3.;
DATALINES;
```

```
JAZZ     252
POP        49
CLASSICAL 59
RAP       21
GOSPEL    44
JAZZ      21
;
ODS RTF;
PROC FREQ DATA=CDS ORDER=FREQ; WEIGHT NUMBER;
  TITLE3 'READ IN SUMMARIZED DATA';
  TABLES CATEGORY;
RUN;
ODS RTF CLOSE;
```

Produces the following table:

CATEGORY	Frequency	Percent	Cumulative Frequency	Cumulative Percent
JAZZ	273	61.21	273	61.21
CLASSICAL	59	13.23	332	74.44
POP	49	10.99	381	85.43
GOSPEL	44	9.87	425	95.29
RAP	21	4.71	446	100.00

Notice that although the data were summarized, there were two observations in the data set for "JAZZ" which were combined into a single category in the table.

Testing Goodness of Fit in a One-Way Table

A goodness-of-fit test of a single population is a test to determine if the distribution of observed frequencies in the sample data closely matches the expected number of occurrences under a hypothetical distribution of the population. The data observations must be independent and each data value can be counted in one and only one category. It is also assumed that the number of observations is fixed. The hypotheses being tested are

Ho: The population follows the hypothesized distribution.
Ha: The population does not follow the hypothesized distribution.

A Chi-Square statistic is calculation and a decision can be made based on the p-value associated with that statistic. A low p-value indicates rejection of the null hypothesis. That is, a low p-value indicates that the data do not follow the hypothesized, or theoretical, distribution.

For example, data for this test comes from Zar (1999), page 465. According to a genetic theory, crossbred pea plants show a 9:3:3:1 ratio of yellow smooth, yellow wrinkled, green smooth, green wrinkled offspring. Out of 250 plants, under the theoretical ratio (distribution) of 9:3:3:1, you would expect about

(9/16)x250=140.625 yellow smooth peas (56.25%)
(3/16)x250=46.875 yellow wrinkled peas (18.75%)
(3/16)x250=46.875 green smooth peas (18.75%)
(1/16)x250=15.625 green wrinkled peas (6.25%)

After growing 250 of these pea plants, you observe that

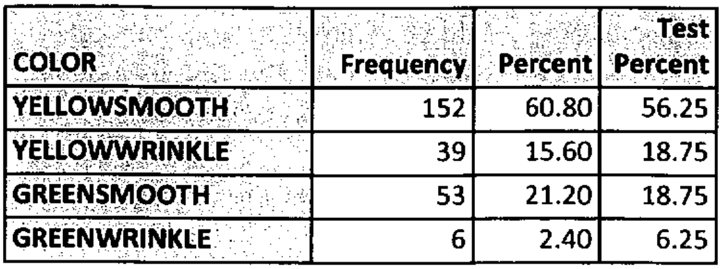
152 have yellow smooth peas
39 have yellow wrinkled peas
53 have green smooth peas
6 have green wrinkled peas

You can perform this analysis using the following SAS program (uses SAS program file **procfreq3**).

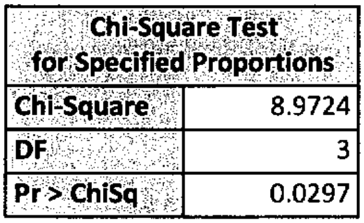
```
DATA GENE;
     INPUT @1 COLOR $13. @15 NUMBER 3.;
DATALINES;
YELLOWSMOOTH  152
YELLOWWRINKLE  39
GREENSMOOTH    53
GREENWRINKLE    6
;
* HYPOTHESIZING A 9:3:3:1 RATIO;
PROC FREQ DATA=GENE ORDER=DATA; WEIGHT NUMBER;
  TITLE3 'GOODNESS OF FIT ANALYSIS';
  TABLES COLOR / CHISQ NOCUM TESTP=(0.5625 0.1875 0.1875 0.0625);
RUN;
```

- The CHISQ requests that a Chi-Square test be performed
- The TESTP=() statement specifies the hypothesized proportions to be tested. (Your could have used the TESTF=() and used expected frequencies instead.)
- The TESTP=() statement specifies the hypothesized proportions to be tested. (Your could have used the TESTF=() and used expected frequencies instead.)
- The NOCUM option suppresses cumulative frequencies
- Use the ORDER=DATA option to cause SAS to displayed data in the same order as they are entered in the input data set.

The result of this analysis is:



COLOR	Frequency	Percent	Test Percent
YELLOWSMOOTH	152	60.80	56.25
YELLOWWRINKLE	39	15.60	18.75
GREENSMOOTH	53	21.20	18.75
GREENWRINKLE	6	2.40	6.25



Chi-Square Test for Specified Proportions	
Chi-Square	8.9724
DF	3
Pr > ChiSq	0.0297

Sample Size = 250

In this case, the p-value for the Chi-Square test is < 0.05 and we reject the null hypothesis and conclude that the peas do not come from a population having the (9:3:3:1) phenotypic ratios.

Analyzing Two-Way Tables

To create a table in PROC FREQ comparing two variables, use the TABLES statement with both variables listed and separated by an asterisk (*). (*i.e.*, A * B), PROC FREQ will produce a cross tabulation table (also called a two-way table).

When you create a two-way cross tabulation, you may want to know the statistics associated with this table. The /CHISQ option in the TABLES statement is used to request that statistics be reported. For example:

```
PROC FREQ; TABLES GENDER*GP/CHISQ;
```

will create a two-way cross tabulation table and will also cause SAS to report a battery of statistics associated with the table.

Test Assumptions: For the Chi-square statistic, the observed data are assumed to be counts of qualitative/categorical data such as hair color, presence of a condition (*i.e.*, a disease or not) *etc.* A cross tabulation table (sometimes called a contingency table) is formed by counting the number of occurrences in a sample across two grouping variables (specified in TABLES). The number of columns in a table is usually denoted by *c* and the number of rows by *r*. Thus, a table is said to have *r x c* "cells." For example, if in a dominate-hand (left-right) by hair color table, (with 5 hair colors used) the table would be referred to as a 2 x 5 table. Two types of tests are commonly associated with an r x c table. They are the test of independence and the test of homogeneity. The hypotheses for the **test of independence** are:

H_o: The variables are independent (no association between the two variables)
H_a: The variables are not independent

Thus, in the "hair" example, the null hypothesis would mean that there is no association between dominant hand and hair color (each hand dominance category has the same distribution of hair color). The alternative hypothesis would mean that left and right-handed people have difference distributions of hair color -- perhaps left-handed people are more likely to be brunette.

Another test that can be performed for a contingency table is a **test of homogeneity**. In this case, the table is built of data from two populations and tests whether the populations come from the same distribution. In this case the hypotheses are:

H_o: The populations are homogeneous.
H_a: The populations are not homogeneous.

Rows (or columns) represent data from different populations, and the other variable represents data observed on the population. The c^2 (Chi-square) test of homogeneity or independence is reported (the tests are mathematically equivalent.) Also included in the output is a likelihood ratio chi-square, Mantel-Hantzel chi-square, phi, contingency coefficient, and Cramer's V. For a 2*2 table, a Fisher's exact test is also performed.

For example, you could create a two-by-two table of GENDER by GP by using the following statements from the SOMEDATA data set (uses SAS program file **procfreq4**).

```
* ASSUMES YOU HAVE A SAS LIBRARY NAMED MYDATA;
ODS RTF;
PROC FREQ DATA=MYDATA.SOMEDATA;
     TABLES GENDER*GP/CHISQ;
TITLE 'Chi Square Analysis of a Contingency Table';
RUN;
* RUN IT AGIN, REQUESTING EXPECTED VALUES;
PROC FREQ DATA=MYDATA.SOMEDATA;
     TABLES GENDER*GP/CHISQ EXPECTED NOROW NOCOL NOPERCENT;
RUN;
ODS RTF CLOSE;
```

The output for the first two-way table in this job (in part) follows:

Table of GENDER by GP				
GENDER	GP(Intervention Group)			
Frequency Percent Row Pct Col Pct	A	B	C	Total
Female	6 12.00 20.00 54.55	16 32.00 53.33 55.17	8 16.00 26.67 80.00	30 60.00
Male	5 10.00 25.00 45.45	13 26.00 65.00 44.83	2 4.00 10.00 20.00	20 40.00
Total	11 22.00	29 58.00	10 20.00	50 100.00

The four numbers in each cell are the frequency, the total percent, percent by row and percent by column. The statistic for this table are given in the next table:

Statistics for Table of GENDER by GP

Statistic	DF	Value	Prob
Chi-Square	2	2.0846	0.3526
Likelihood Ratio Chi-Square	2	2.2433	0.3257
Mantel-Haenszel Chi-Square	1	1.3157	0.2514
Phi Coefficient		0.2042	
Contingency Coefficient		0.2001	
Cramer's V		0.2042	
WARNING: 33% of the cells have expected counts less than 5. Chi-Square may not be a valid test.			

Sample Size = 50

The Chi-Square value is 2.08 with p=.3526. This provides evidence to ***not reject*** the null hypothesis – thus you would conclude that there is no relationship between gender and group. However, notice the warning at the bottom of the table. It tells you that 33% of the cells have expected values of 5 or less, which may make the Chi-Square test invalid. To check this out you look at the version of the table you requested in the second PROC FREQ – this one which requested that the expected values be included in the analysis using

```
TABLES GENDER*GP/CHISQ EXPECTED NOROW NOCOL NOPERCENT;
```

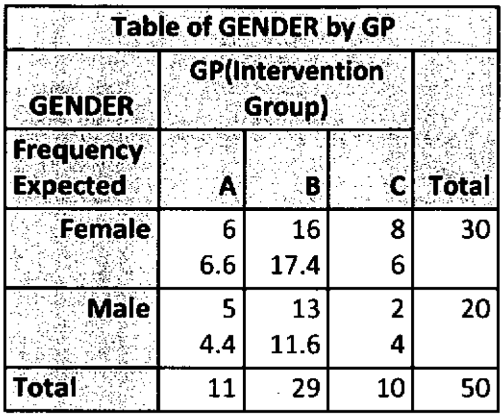


Table of GENDER by GP				
GENDER	GP(Intervention Group)			
Frequency Expected	A	B	C	Total
Female	6 6.6	16 17.4	8 6	30
Male	5 4.4	13 11.6	2 4	20
Total	11	29	10	50

The TABLES statement also requested that ROW, COLUMN and total PERCENTS be excluded from the table. From the resulting table you can see that two of the cells have expected values less than 5 (4.4 and 4). Viewing the expected values can also help you understand why a Chi-Square statistic is significant by observing which observed values depart most from expected values.

EXERCISE: Add FISHERS to the TABLES statement to get Fishers Exact statistic.

```
TABLES GENDER*GP/CHISQ FISHERS EXPECTED NOROW NOCOL NOPERCENT;
```

Fisher's Exact test is often preferred over the Chi-Square when the numbers in the table are small or when the table contains expected values less than 5 (as is true in this example.)

Creating a Contingency Table from Summarized Data

If your data are already summarized into counts, you can use the programming features of SAS to create a dataset appropriate for the analysis (uses SAS program file **procfreq5**). The 2x2 table contains the values 12,15,18, and 3:

12	15
18	3

In the following SAS code, the DO LOOP statements are used to enter this data into a dataset in the proper format for the PROC FREQ statement.

```
DATA;
   DO A = 1 TO 2;
```

```
        DO B = 1 TO 2;
            INPUT WT @@;
            OUTPUT;
        END;
    END;
DATALINES;
12 15
18 3
;
ODS RTF;
PROC FREQ;
    WEIGHT WT;
    TABLES A*B /CHISQ;
    TITLE 'CHI-SQUARE ANALYSIS FOR A 2X2 TABLE';
RUN;
ODS RTF CLOSE;
```

The output for this program follows. The basic table is the same as in the previous example. The Chi-Square statistic is 8.58 (1 df) and p=0.0034. From this evidence you would reject the null hypothesis and conclude that the observations for variable B are influenced by A. For example, looking at the row percentages for A=1, notice that B goes up from 44% to 56%. Whereas when A=2, B goes down from 86% to 14% -- the pattern of B is different across categories of A.

In the 2x2 case, SAS automatically also includes Fisher's Exact Test. Most commonly, the two-sided Fishers p-value (p=.006) would be reported. Fisher's is often preferred over the Chi-Square when the numbers in the table are small or when the table contains expected values less than 5.

The output for these test are given below:

Table of A by B			
A	B		
Frequency Percent Row Pct Col Pct	1	2	Total
1	12 25.00 44.44 40.00	15 31.25 55.56 83.33	27 56.25
2	18 37.50 85.71 60.00	3 6.25 14.29 16.67	21 43.75
Total	30 62.50	18 37.50	48 100.00

Statistics for Table of A by B

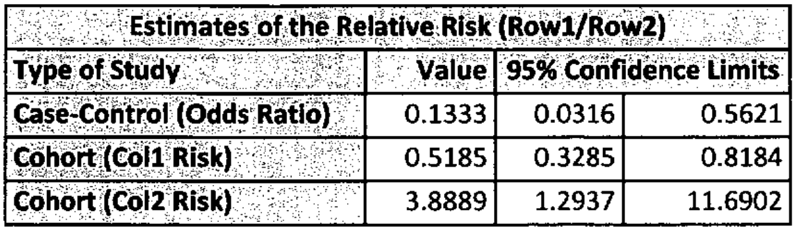
Statistic	DF	Value	Prob
Chi-Square	1	8.5841	0.0034
Likelihood Ratio Chi-Square	1	9.1893	0.0024
Continuity Adj. Chi-Square	1	6.9136	0.0086
Mantel-Haenszel Chi-Square	1	8.4053	0.0037
Phi Coefficient		-0.4229	
Contingency Coefficient		0.3895	
Cramer's V		-0.4229	

tc "Fisher's Exact Test " \f C \l 3Fisher's Exact Test	
Cell (1,1) Frequency (F)	12
Left-sided Pr <= F	0.0036
Right-sided Pr >= F	0.9996
Table Probability (P)	0.0032
Two-sided Pr <= P	0.0061

EXERCISE: Include RELRISK as an option in the TABLE statement:

```
TABLES A*B /CHISQ RELRISK;
```

This yields these additional statistics:



Estimates of the Relative Risk (Row1/Row2)			
Type of Study	Value	95% Confidence Limits	
Case-Control (Odds Ratio)	0.1333	0.0316	0.5621
Cohort (Col1 Risk)	0.5185	0.3285	0.8184
Cohort (Col2 Risk)	3.8889	1.2937	11.6902

It is important to note that the Odds Ratio is based on Row1/Row2. If you switch rows, the Chi-Square statistics are all the same, but the Odds Ratio is the inverse. (1/.1333 = 7.5).

Comparing Independent Group and Paired t-tests

It is not uncommon for researchers to perform an incorrect t-test when comparing two "groups." The correct t-test depends on how the data are observed (the design of the experiment.)

Independent Samples: When data are collected on subjects where subjects are (hopefully randomly) divided into two groups, this is called an independent or parallel study. That is, the subjects in one group (treatment, etc) are different from the subjects in the other group. This data may be analyzed using an independent group t-test (sometimes called an independent samples t-test or parallel test.) This version of the t-test is testing the null hypothesis (two-sided):

H_o: $m_1 = m_2$ (means of the two groups are equal)
H_a: $m_1 {}^1 m_2$ (means are not equal)

Dependent Samples: When data are collected twice on the same subjects (or matched subjects) the proper analysis is a paired t-test (also called a dependent samples t-test). In this case, subjects may be measured in a before – after fashion, or in a design where a treatment is administered for a time, there is a washout period, and another treatment is administered (in random order for each subject). Or, data might be measured on the same individual in two areas such as one treatment in one eye and another treatment for another eye (or leg, or arm, etc). In these cases the measurement of interest is the *difference between the first and second measure*. Thus, the null hypothesis (two-sided) is:

H_o: $\mu_{difference} = 0$ (The average difference is 0)

H_a: $\mu_{difference} \neq 0$ (The average difference is not 0)

Why it makes a difference: Performing an incorrect t-test on your data can cause you to miss a significant difference when one might exist. As an example, consider the data from a paper by Raskin and Unger (1978) where four diabetic patients were used to compare the effects of insulin infusion regimens. One treatment was insulin and somatostatin (IS) and the other treatment was insulin, somatostatin and gulcagon (ISG). Each subject was given each treatment with a period of washout between treatments. The data follow:

Patient	**Treatment**		
Number	**IS**	**ISG**	**Difference**
1	14	17	3
2	6	8	2
3	7	11	4
4	6	9	3
Mean	*8.25*	*11.25*	*3.0*
S.E.M.	*1.9*	*2*	*.40*

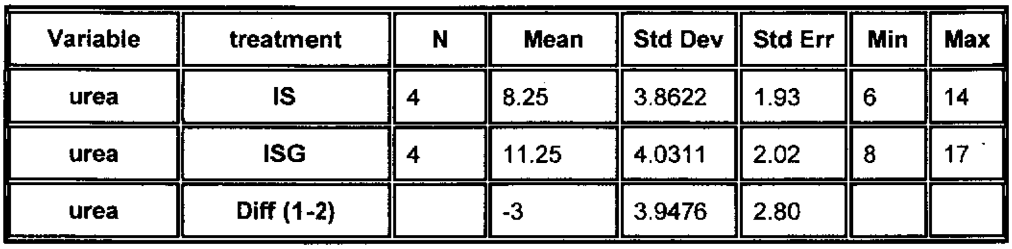
A paper by Thomas Louis (1984) looked at this data using both types of t-tests. The correct version of the t-test to use for this data set is the paired t-test since each patient is observed twice. However, it is all too common for researchers to compare the means 8.25 versus 11.25 using an independent group approach. To see how these approaches differ, consider how the two analyses would be performed in SAS.

Independent group analysis: The code to perform this analysis using an independent group t-test is (uses SAS program file **procttest1**).

```
DATA DIABETIC;
INPUT TREATMENT $ UREA;
DATALINES;
IS  14
IS   6
IS   7
IS   6
ISG 17
ISG  8
ISG 11
ISG  9
;
ODS HTML;
PROC TTEST;
  CLASS TREATMENT;
  VAR UREA;
RUN;
PROC BOXPLOT;
   PLOT UREA*TREATMENT;
RUN;
ODS HTML CLOSE;
```

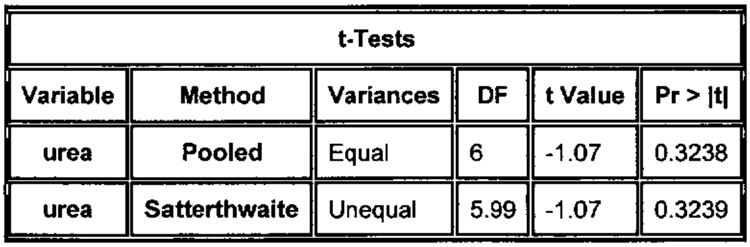
You get the following output (only part of the output is shown here). *(Remember that this is the incorrect t-test to analyze this data):*

The first table shows you that the two means differ by 11.25-8.25 = 3 with a (pooled) standard error of 2.80.



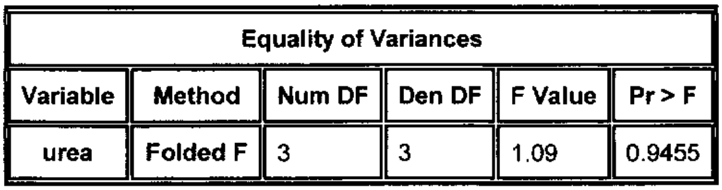
Variable	treatment	N	Mean	Std Dev	Std Err	Min	Max
urea	IS	4	8.25	3.8622	1.93	6	14
urea	ISG	4	11.25	4.0311	2.02	8	17
urea	Diff (1-2)		-3	3.9476	2.80		

Since the "Equality of variances" table below indicates that the variances can be assumed equal (p=.95), you perform the "Pooled/Equal" t-test, which gives a p-value of p=.32. ***(Not a statistically significant result.)***



t-Tests

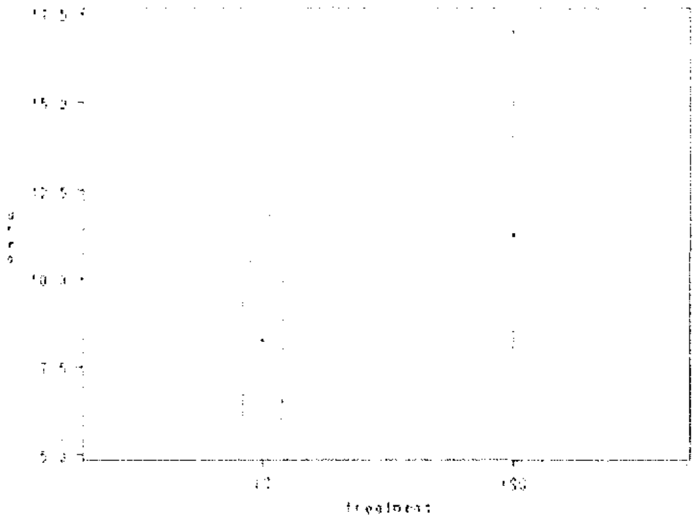
Variable	Method	Variances	DF	t Value	Pr > |t|
urea	Pooled	Equal	6	-1.07	0.3238
urea	Satterthwaite	Unequal	5.99	-1.07	0.3239



Equality of Variances

Variable	Method	Num DF	Den DF	F Value	Pr > F
urea	Folded F	3	3	1.09	0.9455

Furthermore, a comparative box plot shows a lot of overlap between the two groups.



This independent group analysis *is NOT the correct analysis.* This graph, by the way, is also misleading and not appropriate for a paired analysis.

Since the data in this example are paired you should instead do the PAIRED version of the t-test.

Paired t-test analysis: The appropriate analysis for this data is a paired t-test. The calculations for this test can be performed using the following SAS code (uses SAS program file **procttest2**)

```
DATA DIABETIC;
INPUT IS ISG;
DATALINES;
14      17
6       8
7       11
6       9
ODS HTML;
PROC TTEST;
  PAIRED IS*ISG;
RUN;
ODS HTML CLOSE;
```

The (partial) output is as follows. Note that the analysis is performed on the mean of the differences (-4.299) and that the standard error of the difference is 0.41 (much less than the standard error (2.80) in the previous analysis.)

Difference	N	Lower CL Mean	Mean	Upper CL Mean	Lower CL Std Dev	Std Dev	Upper CL Std Dev	Std Err
IS - ISG	4	-4.299	-3	-1.701	0.4625	0.8165	3.0443	0.4082

The paired t-test yields p=0.005, ***which is statistically significant.***

T-Tests			
Difference	**DF**	**t Value**	**Pr > |t|**
IS - ISG	3	-7.35	0.0052

The reason that the paired t-test found significance when the independent t-test *on the same data* did not achieve significance is because the paired analysis is the more correct analysis and therefore it is able to make use of a much smaller standard error (of the mean difference rather than pooled.)

In his paper, Louis explains that to achieve the power of this paired t-test, an independent group t-test (parallel test) would require 14 times as many subjects. Thus, when the model is appropriate, the paired t-test can be a more powerful design to analysis your data. On the other hand, if you use a paired analysis on independent group data you will get incorrect and misleading results. Therefore, carefully consider how your experiment is designed before you select which t-test to perform.

References:

Louis TA, Lavori, PW, Bailer, JC and Polansky, M (1984), "Crossover and Self Controlled Designs in Clinical Research," NEJM, 310:24-31.

Raskin, P, Unger, RH, Hyperglucagonemia and its suppression: importance in the metabolic control of diabetes. NEJM 1978: 299;433-6.

Using PROC ANOVA – One-Way Analysis

A one-way analysis of variance is an extension of the independent group t-test where there are more than two groups.

Assumptions: It is assumed that subjects are randomly assigned to one of 3 or more groups and that the data within each group are normally distributed with equal variances across groups. Sample sizes between groups do not have to be equal, but large differences in sample sizes for the groups may affect the outcome of some multiple comparisons tests.

Test: The hypotheses for the comparison of independent groups are: (k is the number of groups)

H_o: ⍰$_1$ = ⍰$_2$ ... = ⍰$_k$ (means of the all groups are equal)
H_a: ⍰$_i$ ⍰ ⍰$_j$ (means of the two or more groups are not equal)

The test statistic reported is an F test with k-1 and N-k degrees of freedom, where N is the number of subjects. A low p-value for the F-test is evidence to reject the null hypothesis. In other words, there is evidence that at least one pair of means are not equal. For example, suppose you are interested in comparing WEIGHT (gain) across the 4 levels of a GROUP variable, to determine if weight gain of individuals across groups is significantly different.

The following SAS code can perform the test:

```
PROC ANOVA DATA=ANOVA;
CLASS GROUP;
MODEL WEIGHT=GROUP;
TITLE 'Compare WEIGHT across GROUPS';
RUN;
```

GROUP is the "CLASS" or grouping variable (containing four levels), and WEIGHT is the continuous variable, whose means across groups are to be compared. The MODEL statement can be thought of as

```
DEPENDENT VARIABLE = INDEPENDENT VARIABLE(S);
```

where the DEPENDENT variable is the "response" variable, or one you measured, and the independent variable(s) is the observed data. The model statement generally indicated that given the information on the right side of the equal sign you can predict something about the value of the information on the left side of the equal sign. (Under the null hypothesis there is no relationship.)

Since the rejection of the null hypothesis does not specifically tell you which means are different, a **multiple comparison** test is often performed following a significant finding in the One-Way ANOVA. To request multiple comparisons in PROC ANOVA, include a MEANS statement with a multiple comparison option. The syntax for this statement is

```
MEANS SOCIO /testname;
```

where testname is a multiple comparison test. Some of the tests available in SAS include:

```
BON              - Performs Bonferroni t-tests of differences
DUNCAN           - Duncan's multiple range test
SCHEFFE          - Scheffe multiple comparison procedure
SNK              - Student Newman Keuls multiple range test
LSD              - Fisher's Least Significant Difference test
TUKEY            - Tukey's studentized range test
DUNNETT ('x')    - Dunnett's test - compare to a single control
```

You may also specify

```
ALPHA = p   - selects level of significance for comparisons    (default is
0.05)
```

For example, to select the TUKEY test, you would use the statement

```
MEANS GROUP /TUKEY;
```

Graphical comparison

A graphical comparison allows you to visually see the distribution of the groups. If the p-value is low, chances are there will be little overlap between the two or more groups. If the p-value is not low, there will be a fair amount of overlap between all of the groups. A simple graph for this analysis can be created using the PROC PLOT or PROC GPLOT procedure. For example:

```
PROC GPLOT; PLOT GROUP*WEIGHT;
```

will produce a plot showing WEIGHT by group.

Thus, the code for the complete analysis becomes:

```
PROC ANOVA;
CLASS GROUP;
MODEL WEIGHT=GROUP;
MEANS GROUP /TUKEY;
TITLE 'Compare WEIGHT across GROUPS';
PROC GPLOT; PLOT GROUP*WEIGHT;
      RUN;
```

Following is a SAS job that performs a one-way ANOVA and produces a plot.

One-Way ANOVA Example

Suppose you are comparing the time to relief of three headache medicines -- brands 1, 2, and 3. The time to relief data is reported in minutes. For this experiment, 15 subjects were randomly placed on one of the three medicines. Which medicine (if any) is the most effective? The data for this example are as follows:

Brand 1	Brand 2	Brand 3
24.5	28.4	26.1
23.5	34.2	28.3
26.4	29.5	24.3
27.1	32.2	26.2
29.9	30.1	27.8

Notice that SAS expects the data to be entered as two variables, a group and an observation.

Here is the SAS code to analyze these data (uses SAS program file **procanova1**).

```
DATA ACHE;
INPUT BRAND RELIEF;
CARDS;
1 24.5
1 23.5
1 26.4
1 27.1
1 29.9
2 28.4
2 34.2
2 29.5
2 32.2
2 30.1
3 26.1
3 28.3
3 24.3
3 26.2
3 27.8
;
ODS RTF;ODS LISTING CLOSE;
PROC ANOVA DATA=ACHE;
    CLASS BRAND;
    MODEL RELIEF=BRAND;
    MEANS BRAND/TUKEY CLDIFF;
TITLE 'COMPARE RELIEF ACROSS MEDICINES  - ANOVA EXAMPLE';
PROC GPLOT;
       PLOT RELIEF*BRAND;
PROC BOXPLOT;
    PLOT RELIEF*BRAND;
       TITLE 'ANOVA RESULTS';
RUN;
QUIT;
ODS RTF close;
ODS LISTING;
```

Following is the (partial) output for the headache relief study:

ANOVA Procedure
Dependent Variable: Relief

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	66.7720000	33.3860000	7.14	0.0091
Error	12	56.1280000	4.6773333		
Corrected Total	14	122.9000000			

R-Square	Coeff Var	Root MSE	RELIEF Mean
0.543303	7.751664	2.162714	27.90000

Source	DF	Anova SS	Mean Square	F Value	Pr > F
BRAND	2	66.77200000	33.38600000	7.14	0.0091

The initial table in this listing is the Analysis of Variance Table. The most important line to observe in this table is the "Model." At the right of this line is the p-value for the overall ANOVA test. It is listed as "Pr > F" and is p = 0.0091. This tests the overall model to determine if there is a difference in means between BRANDS. In this case, since the p-value is small, you can conclude that there is evidence that there is a statistically significant difference in brands.

Now that you know that there are differences in BRAND, you need to determine where the differences lie. In this case, that comparison is performed by the Tukey Studentized Range comparison (at the alpha = 0.05 level). See the tables below.

The Tukey Grouping table displays those differences. Notice the grouping labels "A" and "B" in this table. There is only one mean associated with the "A" group, and that is brand 2. This indicates that the mean for brand 2 is significantly larger than the means of all other groups. There are two means associated with the "B" group – brands 1 and 3. Since these two means are grouped, it tells you that they were not found to be significantly different.

Tukey's Studentized Range (HSD) Test for RELIEF

Alpha	0.05
Error Degrees of Freedom	12
Error Mean Square	4.677333
Critical Value of Studentized Range	3.77278
Minimum Significant Difference	3.649

Means with the same letter are not significantly different.			
Tukey Grouping	**Mean**	**N**	**BRAND**
A	30.880	5	2
B	26.540	5	3
B			
B	26.280	5	1

Thus, the Tukey comparison concludes that the mean for brand 2 is significantly higher than the means of brands 1 and 3, and that there is no significant difference between brands 1 and 3. Another way to express the differences is to use the CLDIFF option with TUKEY (same results, difference presentation). For example

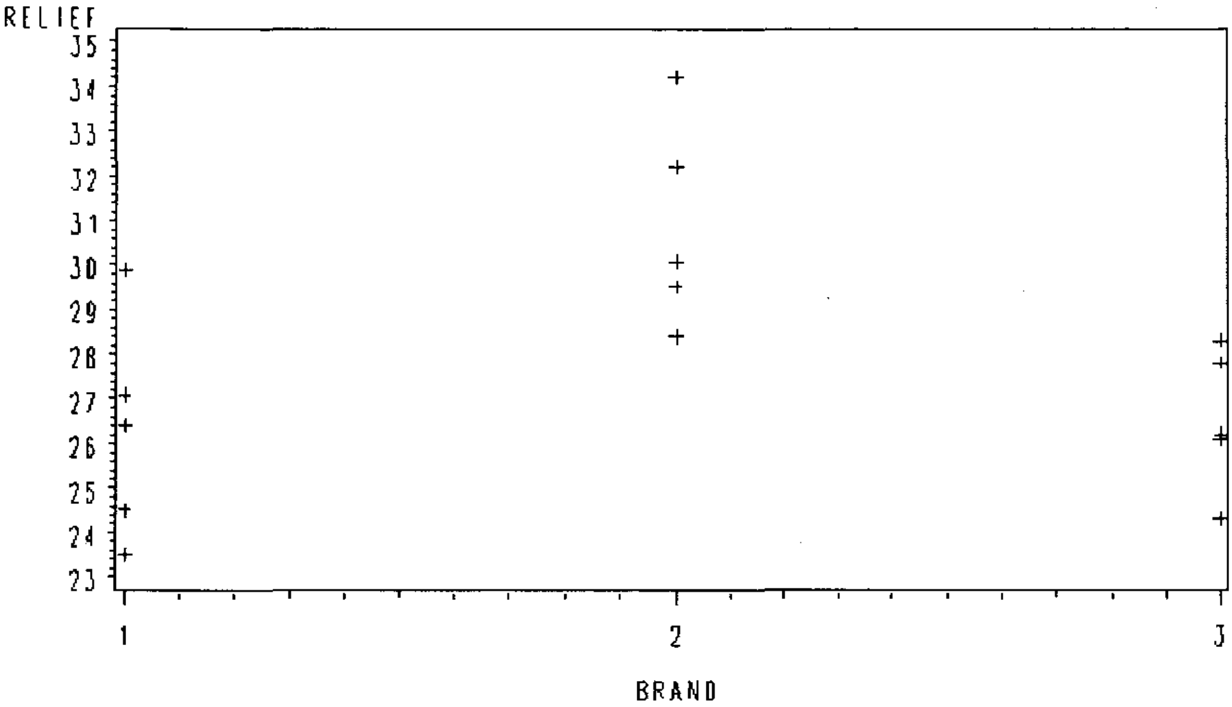
```
MEANS BRAND/TUKEY CLDIFF;
```

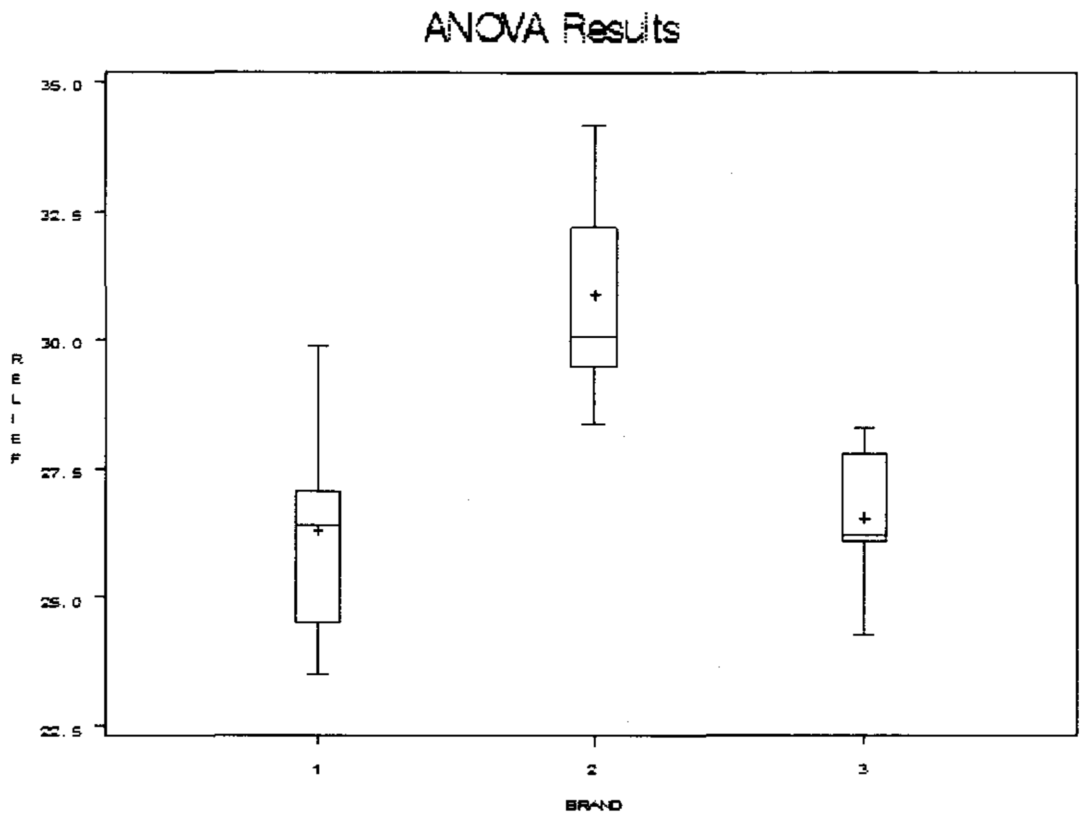
Using this option produces this versions of a comparison table:

Comparisons significant at the 0.05 level are indicated by ***.				
BRAND Comparison	**Difference Between Means**	**Simultaneous 95% Confidence Limits**		
2 - 3	4.340	0.691	7.989	***
2 - 1	4.600	0.951	8.249	***
3 - 2	-4.340	-7.989	-0.691	***
3 - 1	0.260	-3.389	3.909	
1 - 2	-4.600	-8.249	-0.951	***
1 - 3	-0.260	-3.909	3.389	

Visual Comparisons: Two graphs of BRAND by RELIEF shows you the distribution of relief across brands, which visually confirms the ANOVA results. The first is a "dot" plot given by the PROC GPLOT command and shows each data point by group. The second plot is a box and whiskers plot created with PROC BOXPLOT. Note than Brand 2 relief results tend to be longer (higher values) than the levels for brands 1 and 3.

COMPARE RELIEF ACROSS MEDICINES – ANOVA EXAMPLE





Hands-on exercise:

Modify the PROC ANOVA program to perform Scheffe, LSD and Dunnett's test using the following code and compare results.

```
MEANS BRAND/SCHEFFE;
MEANS BRAND/LSD;
MEANS BRAND/DUNNETT ('1');
```

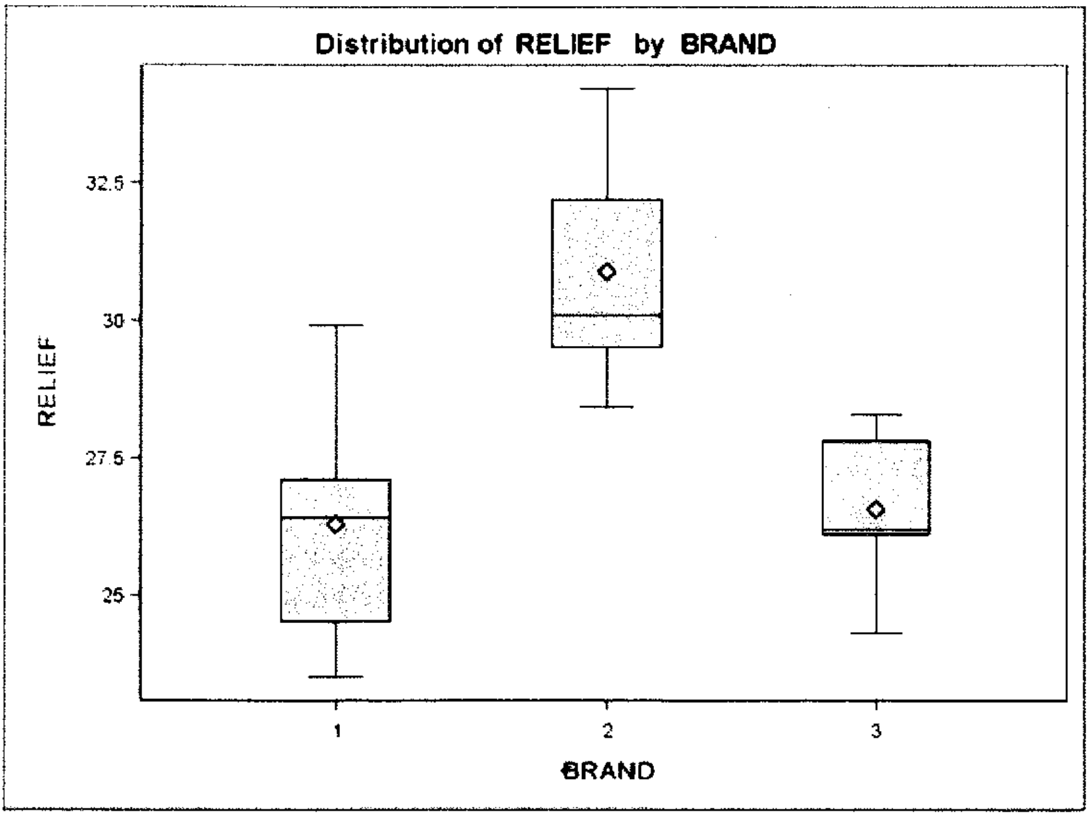
One-Way ANOVA using GLM

PROC GLM will produce essentially the same results as PROC ANOVA with the addition of a few more options. For example, you can include an OUTPUT statement and output residuals that can then be examined (uses SAS program file **procglm1**).

```
ODS RTF; ODS GRAPHICS ON;
PROC GLM DATA=ACHE;
    CLASS BRAND;
    MODEL RELIEF=BRAND;
    MEANS BRAND/TUKEY CLDIFF;
    OUTPUT OUT=FITDATA P=YHAT R=RESID;
```

```
* NOW PLOT THE RESIDUALS;
 PROC GPLOT;
   PLOT RESID*BRAND;
   PLOT RESID*YHAT;
RUN;
ODS RTF CLOSE;
ODS GRAPHICS OFF;
```

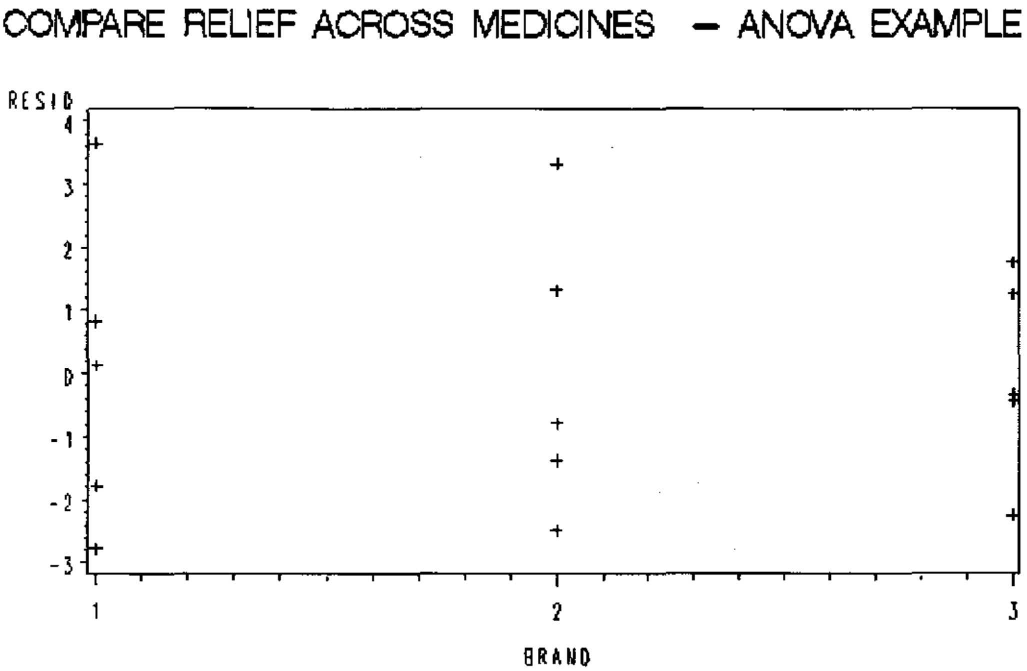
Notice also the statements ODS GRAPHICS ON and ODS GRAPHIS OFF. This produces better looking plots than we were able to get using PROC GPLOT in conjunction with PROC ANOVA. This produces the more detailed box and whiskers plot as show here:



However, there are still a couple of other plots that might be of interest. These are requested using the code

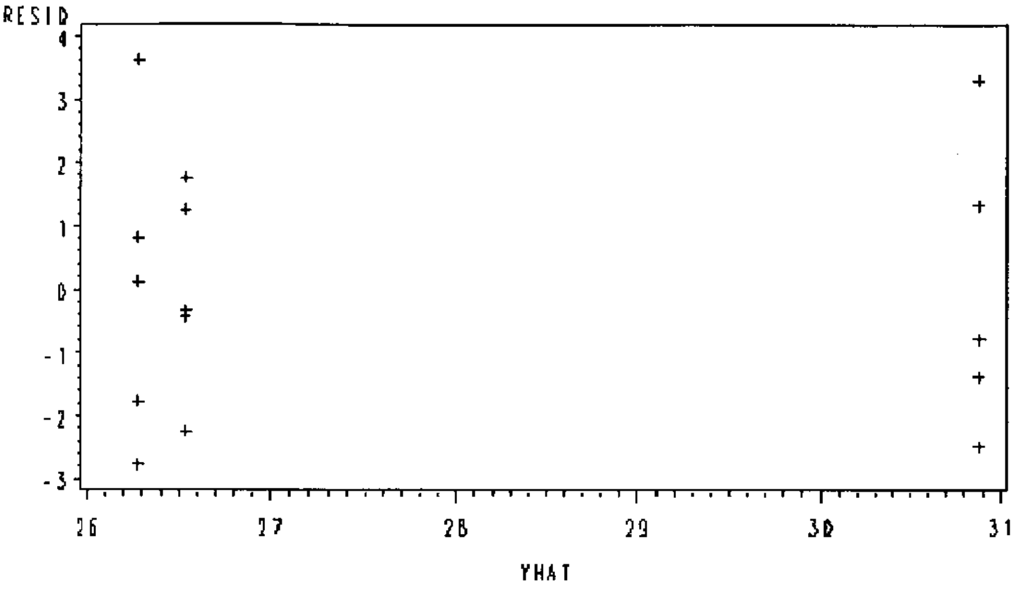
```
PROC GPLOT;
   PLOT RESID*BRAND;
   PLOT RESID*YHAT;
RUN;
```

The resulting plots (below) are an analysis of the residuals. The first plot residuals by brand. Typically, you want the residuals to be randomly scattered by group (which looks okay in this plot)



The second plot looks at residual by YHAT (the estimated RELIEF). You can see three estimates – related to the three brands. For each estimate the residuals are randomly distributed.

COMPARE RELIEF ACROSS MEDICINES — ANOVA EXAMPLE



Repeated Measures Analysis using PROC ANOVA

Repeated Measures are observations taken from the same or related subjects over time or in differing circumstances. Examples would be weight loss or reaction to a drug over time. When there are two repeated measures, the analysis of the data becomes a paired t-test (as discussed earlier). When there are three or more repeated measures, the analysis is a repeated measures analysis of variance. As in the Independent GROUPS ANOVA procedure, you will usually perform the analysis in two steps. First an analysis of variance will determine if there is a difference in means across time. If a difference is evident, then multiple comparisons may be performed to determine where the differences lie.

NOTE: This analysis is also called a within-subjects or treatment-by-subject design. Some call it a "Single-factor experiment having repeated measures on the same element."

Note:

A repeated measures analysis may be performed using PROC ANOVA, PROC GLM, or PROC MIXED. In this discussion, PROC GLM will be used. The syntax used for the other procedures is similar, but each procedure offers a different set of options and capabilities. There are also a number of other Repeated Measures that will discussed in a different tutorial.

The hypotheses being tested with a repeated measures ANOVA is:

Ho: There is no difference among means of the groups (repeated measures).
Ha: There is a difference among means of the groups.

The data in the following example are repeated measures of reaction times of five persons after being treated with four drugs in randomized order. (This type of data may come from a crossover experimental design.) The data are as follows:

Subj	Drug1	Drug2	Drug3	Drug4
1	31	29	17	35
2	15	17	11	23
3	25	21	19	31
4	35	35	21	45
5	27	27	15	31

If you data are in this form, you must first restructure the data into this format:

Subj	Drug	Time
1	1	31
1	2	29
1	3	17
etc. . .		
5	1	7
5	2	27
5	3	15
5	4	31

The following code restructures the data into the format needed for the analysis. Note that the DRUG variable here goes from 1 to 4 **"DO DRUG =1 to 4;"** representing the number of repeated measures in the data.

```
DATA STUDY;
      SUBJ+1;
      DO DRUG =1 to 4;
      INPUT OBS @;
            OUTPUT;
            END;
DATALINES;
31 29 17 35
15 17 11 23
25 21 19 31
35 35 21 45
27 27 15 31
;
```

This program creates a data set called STUDY whose data values are in the form of the second table above. To perform a One-way ANOVA, add the following lines to this code: (use SAS pogram file **procglm2).**

```
PROC GLM DATA=STUDY;
```

```
        CLASS SUBJ DRUG;
        MODEL OBS= SUBJ DRUG;
        MEANS DRUG/TUKEY;
              TITLE 'Repeated Measures ANOVA;
RUN;
```

The (partial) output for this analysis follows:

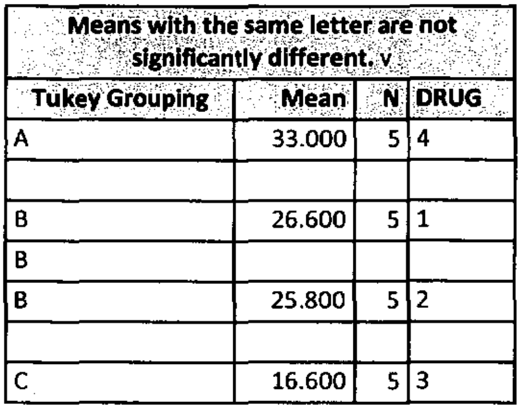
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	1331.800000	190.257143	25.03	<.0001
Error	12	91.200000	7.600000		
Corrected Total	19	1423.000000			

R-Square	Coeff Var	Root MSE	OBS Mean
0.935910	10.81102	2.756810	25.50000

Source	DF	Type III SS	Mean Square	F Value	Pr > F
SUBJ	4	648.0000000	162.0000000	21.32	<.0001
DRUG	3	683.8000000	227.9333333	29.99	<.0001

The "Type III SS" analysis of variance table (DRUG line,) reports a p-value of p < 0.0001. This gives evidence to reject the null hypothesis that there is no difference in the drugs. Since there is a difference in drugs, a multiple comparison test is performed. The results of that test age presented in the next few tables:

Alpha	0.05
Error Degrees of Freedom	12
Error Mean Square	7.6
Critical Value of Studentized Range	4.19852
Minimum Significant Difference	5.1763



Means with the same letter are not significantly different. v			
Tukey Grouping	Mean	N	DRUG
A	33.000	5	4
B	26.600	5	1
B			
B	25.800	5	2
C	16.600	5	3

The Tukey multiple comparison test for DRUGS indicates that the time to relief for DRUG 3 is significantly lower than for all other drugs. There is no statistical difference between drugs 2 and 1, and DRUG 4 has the highest time to relief for all drugs tested.

ALTERNATIVE METHOD: If your data are already in the form of the second table above, your code would be as follows (uses SAS program file **proccglm2**).
The results are the same.

```
DATA STUDY;
INPUT SUBJ DRUG OBS;
DATALINES;
1       1       31
1       2       29
1       3       17
1       4       35
2       1       15
...etc
5       3       15
5       4       31
;
run;
ODS RTF; ODS GRAPHICS ON;
PROC GLM DATA=STUDY;
        CLASS SUBJ DRUG;
        MODEL OBS= SUBJ DRUG;
        MEANS DRUG/TUKEY;
        TITLE 'Repeated Measures ANOVA';
RUN;
ODS RTF CLOSE; ODS GRAPHICS OFF;
```

This discussion only covered the case of a "one-way repeated measures analysis." There are a number of more complex repeated measures analysis of variance designs that will be discussed later.

Controlling SAS output using ODS

ODS, or Output Delivery System, is a method within SAS of controlling the output from SAS Procedures. It was designed to overcome the limitations of traditional SAS output. ODS began development with SAS Version 8 and is now been fully implemented in Version 9.2 .

ODS can be used to output results using several types of format including:

- Basic SAS output (Listing)
- Output in HTML format (html)
- Output to Acrobat (pdf)
- Output as Rich Text Format (rtf) (Can be read by Microsoft Word)
- Output to Postscript

ODS can also be used to

- Output data sets
- Output Graphs associated with procedures

Not every option available in ODS is covered in this introduction to ODS. However, most SAS users should find that the information discussed will cover most output needs.

A Simple ODS Example

This example shows how you can utilize ODS quickly and simply. Following this example, we will look at options for using ODS that will give you more control over the output.

Suppose you want to create output to be saved in Microsoft Word.

Using the following code:

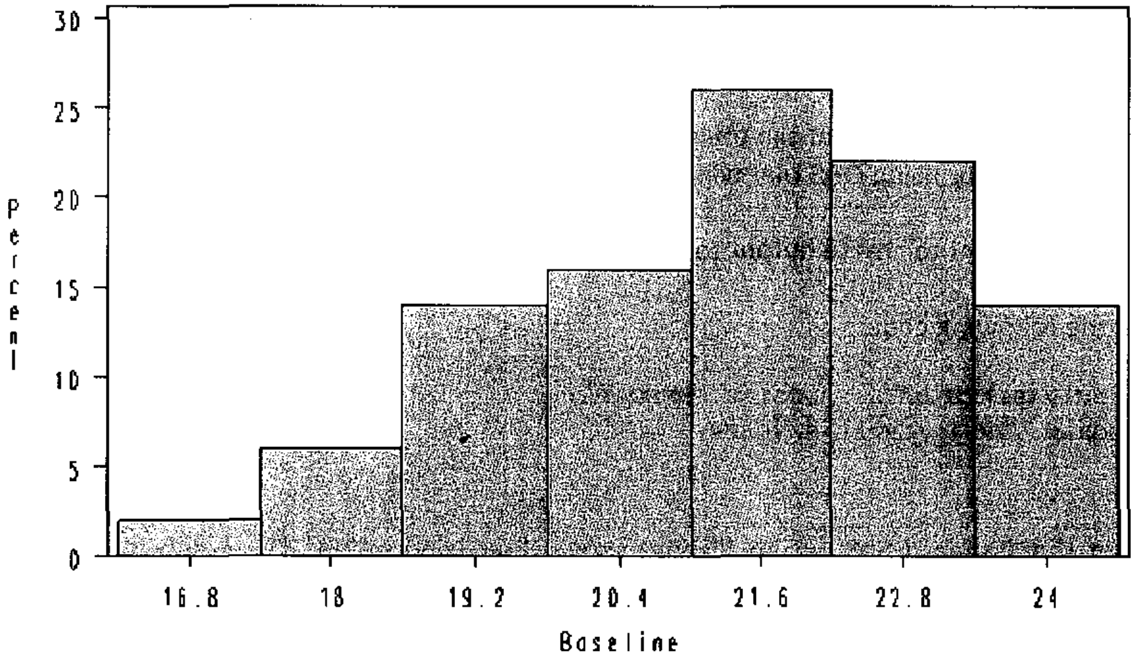
```
PROC UNIVARIATE DATA=SASDATA2.SOMEDATA;
HISTOGRAM TIME1 /CFILL=SKYBLUE;
RUN;
```

you get the standard SAS output (listing) shown (partially) below. The graph is in a separate window, and colors available for the graph are limited.

```
                              The UNIVARIATE Procedure
                             Variable:  ID  (ID Number)

                                       Moments

          N                             50    Sum Weights                50
```

```
            Mean                    374.22    Sum Observations          18711
            Std Deviation       167.498314    Variance             28055.6853
            Skewness            -0.2716195    Kurtosis             -1.3081517
            Uncorrected SS         8376759    Corrected SS         1374728.58
            Coeff Variation     44.7593165    Std Error Mean       23.6878388


                                  Basic Statistical Measures

                     Location                         Variability

                 Mean     374.2200     Std Deviation              167.49831
                 Median   403.5000     Variance                       28056
                 Mode       .          Range                      503.00000
                                       Interquartile Range        310.00000

... etc (output truncated)
```

```
The graph output is:
```

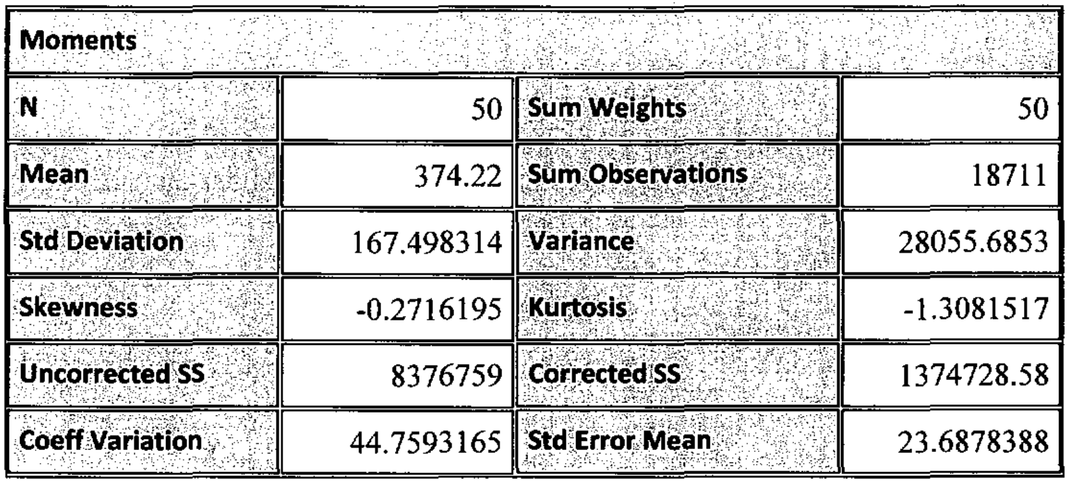


By adding a simple ODS command to the code, as shown here:

```
ODS RTF;
PROC UNIVARIATE DATA=SASDATA2.SOMEDATA;
HISTOGRAM TIME1 /CFILL=SKYBLUE;
RUN;
```

```
ODS RTF CLOSE;
```

The **ODS RTF;** command turns on the ODS output in "Rich Text Format" which is compatible with Word. The **ODS RTF CLOSE;** command turns off the ODS output. The output is opened in a Microsoft Word document as show here (partial output)



Moments			
N	50	Sum Weights	50
Mean	374.22	Sum Observations	18711
Std Deviation	167.498314	Variance	28055.6853
Skewness	-0.2716195	Kurtosis	-1.3081517
Uncorrected SS	8376759	Corrected SS	1374728.58
Coeff Variation	44.7593165	Std Error Mean	23.6878388

This output is in standard Word tables for text output, and also includes the previous graph in the same Word file.

Note that SAS initially saves the output in to an ".RTF" file. You can resave the output as a ".DOC" file to make it into a standard Word document by selecting (in Word) File/Save As... and choosing the "Files of Type" as ".doc".)

Defining the ODS Output Type and Destination

The syntax of the ODS command is

```
ODS OUTPUT-FORMAT <OPTIONS>;
```

The default SAS output location is the normal "listing" which appears in the Output window. This output can be turned on using the command:

```
ODS LISTING;
```

and turned off using

```
ODS LISTING CLOSE;
```

To turn on any other output type simply use the command

```
      ODS OUTPUT-FORMAT;
```

For example

```
      ODS HTML;
```

The turn the output off use the CLOSE option:

```
      ODS HTML CLOSE;
```

Usually, when you are using one of the other output types you may want to turn off the default listing first, output to your selected type, then turn the listing option back on. For example:

```
      ODS LISTING CLOSE;
      ODS RTF;
          * SOME PROCEDURES…;
      ODS RTF CLOSE;
      ODS LISTING;
```

The RTF listing appears in the SAS Results Viewer and in the case of RTF output, SAS also prompts you for a filename which allows you to store the information into a specific RTF file

Output ODS to a Specific File

One of the most useful <OPTIONS> is the ability to direct the output directly into a specific file. The following examples show how you can "open" the output into a specific type of file.

```
      ODS HTML BODY='HTML-FILE-PATHNAME.HTML';
      ODS PDF FILE='PDF-FILE-PATHNAME.PDF';
      ODS PS FILE='PS-FILE-PATHNAME.PS';
```

Notice that the HTML output uses BODY= and the others use FILE=. HTML output can also be output to frames, but that is not discussed here. (Technically, the FILE= option also works for HTML, but if you decide on using other HTML options (such as frames) it is a good idea to stick with the BODY=option.)

For example, to send data to an HTML file named C:\RESEARCH\OUTPUT.HTM you could use the syntax:

```
      ODS HTML BODY= 'C:\RESEARCH\OUTPUT.HTM';
```

after running whatever SAS PROCS whose output you want to capture in the file, you turn off this ODS output using the command

```
ODS HTML CLOSE;
```

In a program you would typically first turn off the default listing, open the ODS output, run procedures, close ODS and reopen the standard listing:

```
ODS LISTING CLOSE;
ODS HTML FILE= 'C:\RESEARCH\OUTPUT.HTM';
      PROC PRINT;
      PROC MEANS;
      *AND SO ON...;
ODS HTML CLOSE;
ODS LISTING;
```

If you leave off the destination as in

```
ODS PDF;
*. . .some procedures;
ODS CLOSE PDF;

or

ODS RTF;
ODS CLOSE RTF;
```

You will asked to enter an output filename (or the results will go into the SAS Results viewer).

Note: All output from any procedure that prints to the output window or the first file that exists between ODS HTML...; and OLDS HTML CLOSE; statements will be sent to that ODS destination. So it is generally best to surround the procedure that is going to produce output you would like to redirect with the opening and closing ODS statements.