The GLM Procedure

When comparisons among two or more population means are to be made, and each population is normally distributed, an Analysis Of Variance (ANOVA) can be done to analyze the sample means and draw inferences about the populations. The hypotheses and test statistic for this test are:

Hypothesis	**Test Statistic**	**Reject H_0 if**
H_0: $\mu_1 = \mu_2 = \ldots = \mu_t$ H_1: at least one μ_i is different	$F = \dfrac{\frac{1}{t-1}\sum_{i=1}^{t}(\bar{y}_{i\bullet} - \bar{y}_{\bullet\bullet})^2}{\frac{1}{N-t}\sum_{i=1}^{t}\sum_{j=1}^{n_i}(y_{ij} - \bar{y}_{i\bullet})^2} = \dfrac{MSTrt}{MSE}$	$F \geq F_{\alpha,t-1,N-t}$ where n_i is the sample size for the ith group, N is the grand sample size; $N = \sum_{i=1}^{t} n_i$

Note: When t=2, the independent t-test statistic yields the same result as the ANOVA F-test: that is, $t^2_{N-t} = F_{1,N-t}$. ANOVA procedures are recommended when the t populations the samples are drawn from are normally distributed.

The GLM (General Linear Model) procedure is used to do this. The syntax of the procedure is this:

```
PROC GLM DATA=tablename <options> ;
CLASS group ;
MODEL response = group ;
MEANS group / <options> ;
```

PROC GLM statement options include

PLOTS= NONE | DIAGNOSTICS RESIDUALS MEANPLOT

ODS Graphics can be requested in the PROC GLM statement. If ODS Graphics are enabled, the MEANPLOT is the default image. If ODS Graphics are enabled (ODS GRAPHICS ON; earlier in the program), one can suppress graphics by including PLOTS=NONE as the option. If a list of ODS Graphics are to be included, one should enclose the list in parenthesis, such as PLOTS = (DIAGNOSTICS RESIDUALS) . DIAGNOSTICS and RESIDUALS are by default assembled as a panel of graphs. One can produce each graph individually by "unpacking" them. The UNPACK option is a global plot option. It appears to the left of the equal sign and is enclosed in parenthesis such as, PLOTS (UNPACK) = RESIDUALS .

CLASS statement — This statement names the variable that identifies the populations to be compared. Class variables can be either numeric or character and must have two or more different values.

MODEL statement — The MODEL statement is order important in that response variables are always listed on the left side of the equal sign, and the class variables always appear on the right. One can list several response variables in a single MODEL statement, such as
response1 response2 . . . responsen = group ;
For each response variable, an ANOVA table will be produced. Only one MODEL statement can be used in a block of SAS/GLM programming. When ODS Graphics are enabled, a MEANPLOT with the ANOVA F-test results inset will be generated unless you suppress it using PLOTS=NONE in the PROC GLM statement.

MEANS statement — This statement will produce the sample means for the class variable listed in this statement. Multiple comparisons procedures can be specified in the options for this statement. LSD, TUKEY, and SNK identify some of the available methods. LINES, CLDIFF, CLM identify comparison and confidence interval calculations that can also be done.
An additional MEANPLOT will be produced by default when using a MEANS statement.

BY and WHERE statements can be added to this procedure as needed. The GLM procedure has many, many facets and hence, many more statements and options available. Only a brief introduction to the procedure is presented here.

Objective 1: Compare the response variable means for three populations (or groups) denoted by A, B, and C. Independent samples are drawn from each of the populations. Compute an ANOVA table and the sample means for each of the three groups. Generate both the Listing Output and the HTML Output.

```
DM 'LOG; CLEAR; ODSRESULTS; CLEAR; ';

DATA one;
INPUT group $ response @@;
DATALINES;
A 15 A 24 A 23 A 24
B 17 B 13 B 15 B 18
C 21 C 25 C 27 C 23
PROC GLM DATA=one;
CLASS group ;
MODEL response = group;
MEANS group ;
TITLE 'Objective 1';
RUN;

QUIT;
```

CLASS group;

```
                                Objective 1                                1

                              The GLM Procedure

                          Class Level Information

                       Class         Levels    Values

                       group              3    A B C


                  Number of Observations Read          12
                  Number of Observations Used          12
```

MODEL response = group ;

```
                                Objective 1                                2

                              The GLM Procedure

Dependent Variable: response

                                      Sum of
Source                          DF       Squares    Mean Square   F Value   Pr > F

Model                            2    143.1666667     71.5833333      7.02   0.0145   ← As MODEL becomes more complex do NOT read this line.

Error                            9     91.7500000     10.1944444                       

Corrected Total                 11    234.9166667


           R-Square     Coeff Var      Root MSE    response Mean

           0.609436      15.63857      3.192874         20.41667


Source                          DF      Type I SS    Mean Square   F Value   Pr > F

group                            2    143.1666667     71.5833333      7.02   0.0145


Source                          DF    Type III SS    Mean Square   F Value   Pr > F

group                            2    143.1666667     71.5833333      7.02   0.0145   ←Read test result here.
```

There is a sig difference among the three group means (α=0.05, $F_{2,9}$ = 7.02, p = 0.0145).

MEANS group;

Objective 1 3

The GLM Procedure

Level of group	N	response Mean	response Std Dev
A	4	21.5000000	4.35889894
B	4	15.7500000	2.21735578
C	4	24.0000000	2.58198890
	n_i	ybar	s_i

CLASS group;

Objective 1

The GLM Procedure

Class Level Information

Class	Levels	Values
group	3	A B C

Number of Observations Read	12
Number of Observations Used	12

MODEL response = group ;

Objective 1

The GLM Procedure

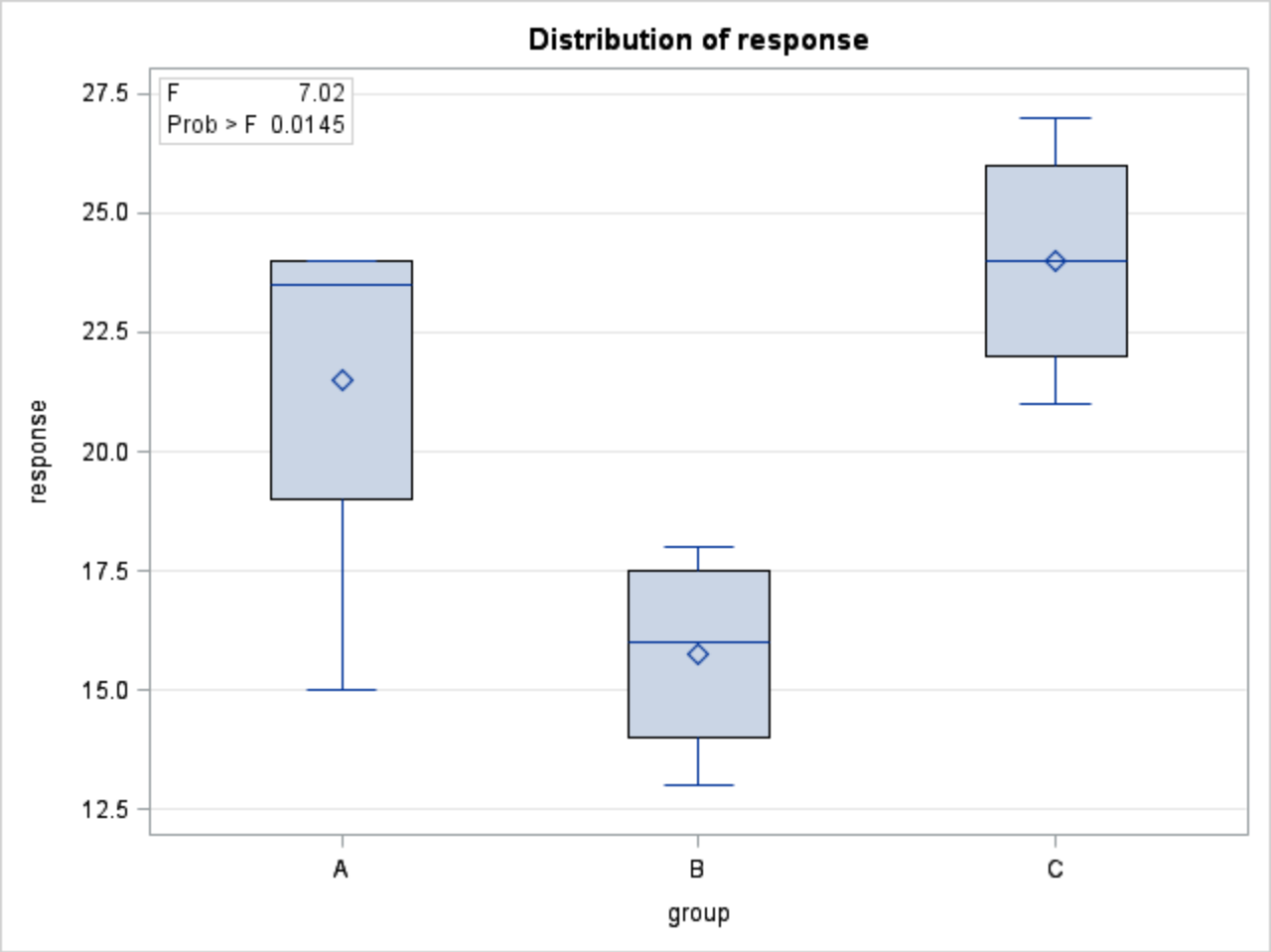
Dependent Variable: response

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	143.1666667	71.5833333	7.02	0.0145
Error	9	91.7500000	10.1944444		
Corrected Total	11	234.9166667			

R-Square	Coeff Var	Root MSE	response Mean
0.609436	15.63857	3.192874	20.41667

Source	DF	Type I SS	Mean Square	F Value	Pr > F
group	2	143.1666667	71.5833333	7.02	0.0145

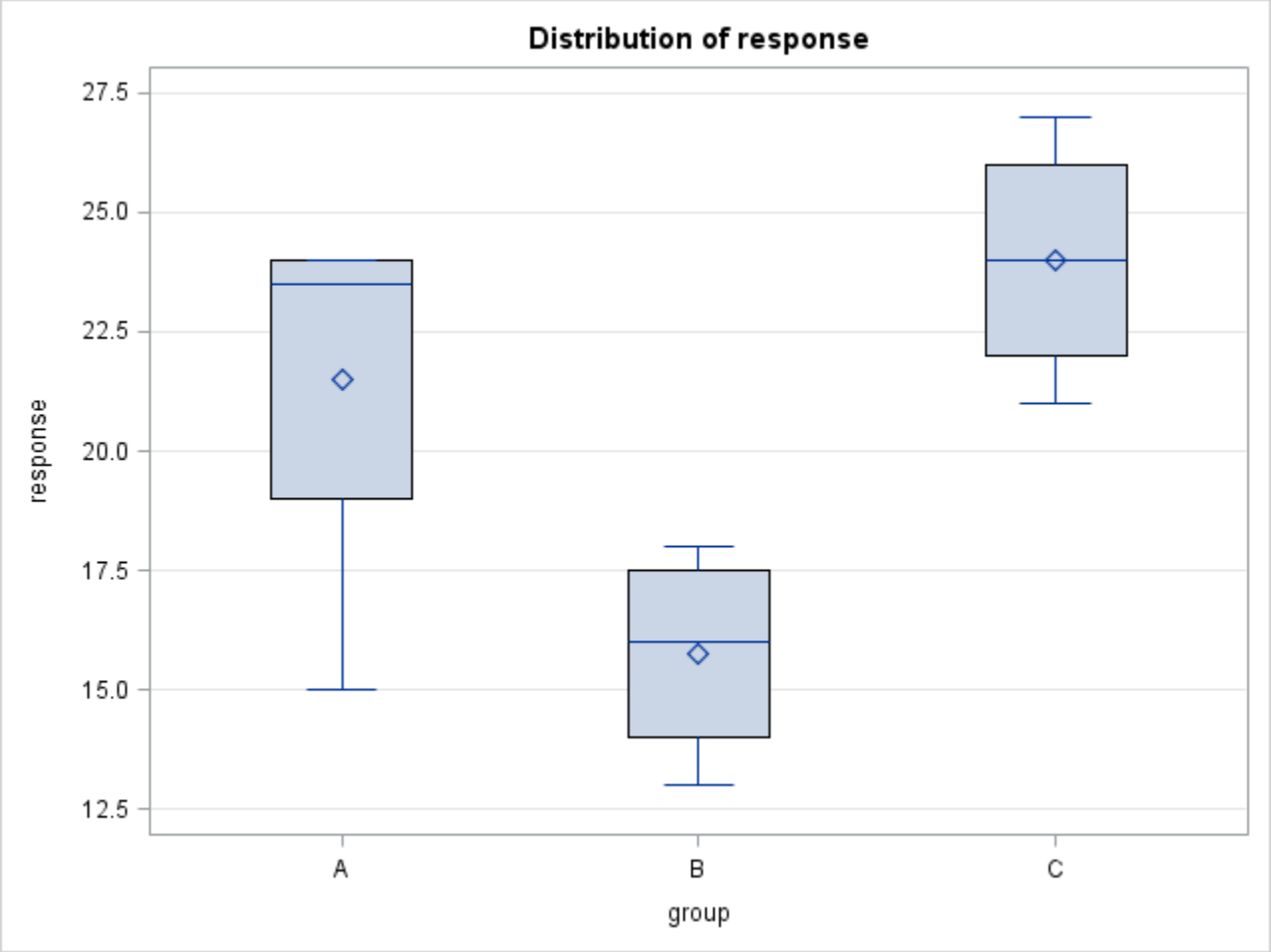
Source	DF	Type III SS	Mean Square	F Value	Pr > F
group	2	143.1666667	71.5833333	7.02	0.0145


Distribution of response
F 7.02
Prob > F 0.0145
27.5
25.0
22.5
20.0
17.5
15.0
12.5
response
A
B
C
group

MEANS group;

Objective 1

The GLM Procedure



Level of group	N	response	
		Mean	Std Dev
A	4	21.5000000	4.35889894
B	4	15.7500000	2.21735578
C	4	24.0000000	2.58198890

Objective 2: Modify the HTML Output with the following options on the PROC GLM statement:

a. suppressing the meanplot
PLOTS = NONE

b. produce residual and diagnostic plots first as a panel graph, and second as individual graphs
PLOTS = (RESIDUALS DIAGNOSTICS)

PLOTS(UNPACK) = (RESIDUALS DIAGNOSTICS)

The NPAR1WAY Procedure

When comparing testing the equality of two or more population means, we can use the ANOVA method when the populations are normally distributed. What if they are not normally distributed? Since ANOVA procedures no longer apply, a class of tests called nonparametric tests could be used. Nonparametric methods do not assume a particular type of distribution. Many nonparametric rank the data and base comparisons on functions of these ranks. Two nonparametric procedures are outlined here.

Method involved with the following two test procedures:

1. Combine the samples.
2. Rank the observations from smallest to largest. Assign the average rank to scores that are tied.
3. Sum the ranks from each of the t samples. Call this sum, R_i for i = 1, 2, . . . , t.

Hypotheses	**Test Statistic**	**Reject H_0 if**	**Comment**
H_0: $\mu_1 = \mu_2 = \ldots = \mu_t$ H_1: at least one μ_i is different	$\chi^2 = \frac{12}{N(N+1)}\sum_{i=1}^{t}\frac{R_i^2}{n_i} - 3(N+1)$ where $N = \sum_{i=1}^{t} n_i$	$\chi^2 \geq \chi^2_{\alpha,t-1}$	This is referred to as the **Kruskal-Wallis** test.
H_0: $\mu_1 = \mu_2$ H_1: $\mu_1 \neq \mu_2$	$U = n_1 n_2 + \frac{n_1(n_1+1)}{2} - R_1$ $E(U) = \frac{n_1 n_2}{2}$ and $\sigma_U = \sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}}$ $z = \frac{U - E(U)}{\sigma_u}$	$\lvert z \rvert \geq z_{\alpha/2}$	This is referred to as the **Wilcoxon Rank Sum Test** or the Mann-Whitney test. This is equivalent to the Kruskal-Wallis Test

The NPAR1WAY procedure performs an analysis on ranks, and it can compute several statistics certain based on rank scores of a response variable. Only one class variable can be specified, hence the "1WAY" part of the procedure name. The NPAR1WAY procedure is used for independent samples only. When samples are dependent or paired, the UNIVARIATE procedure should be used. The UNIVARIATE procedure performs a sign test for paired samples, and the Wilcoxon Signed Rank test for independent samples. There are other nonparametric test options that NPAR1WAY can compute, and these are listed below. Introductory statistics courses generally include the Wilcoxon test.

The syntax of the NPAR1WAY procedure is:

```
PROC   NPAR1WAY   DATA=setname  <options> ;
CLASS  variable ;
VAR variables ;
RUN;
```

PROC NPAR1WAY, CLASS, and VAR are required statements. The BY and WHERE statements are optional statements that can also be used.

The options for the PROC NPAR1WAY statement are:

ANOVA	performs a standard analysis of variance on the raw data
EDF	calculates statistics based on the empirical distribution function. These always include the Kolmogorov-Smirnov and Cramer-von Mises statistics, and if there are only two classification levels, the Kuiper statistic.
MISSING	interprets missing values of the CLASS variable as a valid class level
MEDIAN	performs an analysis of the median scores. The median score is 1 for points above the median, and 0 otherwise. For two samples, this produces the median test. For more that two samples, this produces the Brown-Mood test.
SAVAGE	analyzes Savage scores. These are the expected order statistics for the exponential distribution, with 1 subtracted to center the scores around 0. This test is appropriate for comparing groups of data with exponential distributions.
VW	analyzes Van der Waerden scores. These are approximate normal scores derived by applying the inverse normal distribution function to the fractional ranks.
WILCOXON	performs an analysis of the ranks of the data or the Wilcoxon scores. For two levels, this is the same as the Wilcoxon rank-sum test. For any number of levels, this is a Kruskal-Wallis test. For the two sample cases, the procedure uses a continuity correction.
PLOTS = *list*	Each of the above test methods may have an associated plot. Select the plot type that "goes with" the test procedure selected. For example, for a Wilcoxon test the PLOTS option should read PLOTS=WILCOXONBOXPLOT. If all tests are performed, one would select PLOTS=ALL, and plots can be suppressed by PLOTS=NONE.

CLASS Statement

The CLASS statement, which is required, names one and only one classification variable. This variable identifies groups in the data. Class variables can be character or numeric. Class variables must have two or more levels.

VAR Statement

The VAR statement names the response or dependent variables to be analyzed. If the VAR statement is omitted, the procedure analyzes all numeric variables in the data set (except for the CLASS variable, if it is numeric).

Objective 3: Environmental engineers were interested in determining whether a cleanup project on a nearby lake was effective. Prior to initiation of the project, 12 water samples had been obtained at random from the lake and analyzed for the amount of dissolved oxygen, all measurements were obtained at the 2 pm peak period. The before and after data appear below. These samples are regarded as independent in this study.

Before Cleanup		After Cleanup	
11.0	11.6	10.2	10.8
11.2	11.7	10.3	10.8
11.2	11.8	10.4	10.9
11.2	11.9	10.6	11.1
11.4	11.9	10.6	11.1
11.5	12.1	10.7	11.3

A nonparametric alternative for comparing the mean dissolved oxygen amounts before and after cleanup is used. The SAS code for the analysis appears below. The Listing Output is included and is annotated.

```
DM 'LOG; CLEAR; ODSRESULTS; CLEAR; ';

DATA two;
INPUT  time $ oxygen @@;
DATALINES;
B 11.0    B 11.6    A 10.2    A 10.8
B 11.2    B 11.7    A 10.3    A 10.8
B 11.2    B 11.8    A 10.4    A 10.9
B 11.2    B 11.9    A 10.6    A 11.1
B 11.4    B 11.9    A 10.6    A 11.1
B 11.5    B 12.1    A 10.7    A 11.3
;
PROC NPAR1WAY DATA = two WILCOXON PLOTS=WILCOXONBOXPLOT;
CLASS  time;
VAR oxygen;
TITLE 'Objective 3';
RUN;
ODS HTML CLOSE ;
ODS GRAPHICS OFF ;

QUIT;
```

If you specify no options on the PROC NPAR1WAY statement you will get several pages of output as the procedure will run all possible non-parametric tests. Here only the Wilcoxon tests are overviewed, so select only that option.

This is the LISTING Output for the NPAR1WAY procedure.

Objective 3

The NPAR1WAY Procedure

Wilcoxon Scores (Rank Sums) for Variable oxygen
Classified by Variable time

time	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
B	12	216.0	150.0	17.290359	18.0
A	12	84.0	150.0	17.290359	7.0

Average scores were used for ties.

Wilcoxon Two-Sample Test

Statistic	216.0000
Normal Approximation	
Z	3.7882
One-Sided Pr > Z	<.0001
Two-Sided Pr > |Z|	0.0002
t Approximation	
One-Sided Pr > Z	0.0005
Two-Sided Pr > |Z|	0.0010

Z includes a continuity correction of 0.5.

Kruskal-Wallis Test

Chi-Square	14.5707
DF	1
Pr > Chi-Square	0.0001

R_B = 216.0 and R_A = 84.0 are the rank sum values in step 2 of the method overview.

Before and After times do differ (α=0.01, z = 3.7882, p = 0.0002
or χ^2=14.5707, p =.0001)

Wilcoxon Two-Sample Test information will only print if t = 2. If t > 2, then only the Kruskal-Wallis Test will print.

Only interpret the two-sided test since that is all that was covered in the brief methods review in the notes.

This is the HTML Output for the same procedure.

Objective 3

The NPAR1WAY Procedure

Wilcoxon Scores (Rank Sums) for Variable oxygen Classified by Variable time

time	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
B	12	216.0	150.0	17.290359	18.0
A	12	84.0	150.0	17.290359	7.0

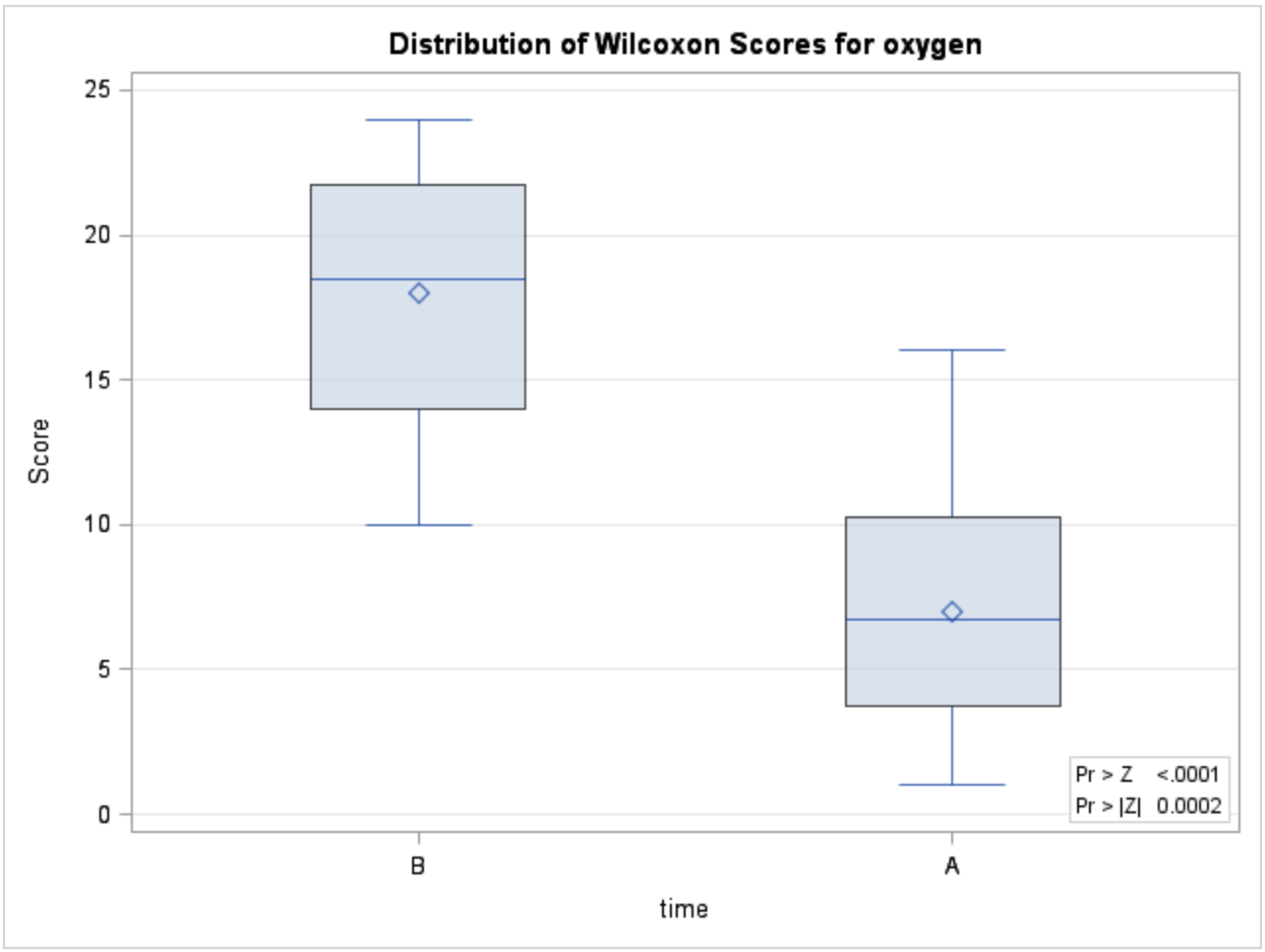
Average scores were used for ties.

Wilcoxon Two-Sample Test	
Statistic	216.0000
Normal Approximation	
Z	3.7882
One-Sided Pr > Z	<.0001
Two-Sided Pr > |Z|	0.0002
t Approximation	
One-Sided Pr > Z	0.0005
Two-Sided Pr > |Z|	0.0010

Z includes a continuity correction of 0.5.

Kruskal-Wallis Test	
Chi-Square	14.5707
DF	1
Pr > Chi-Square	0.0001

PLOTS=WILCOXONBOXPLOT



Ranking Items in a Data Set

While the SORT procedure can arrange the data in order, it does not assign a place or rank to the sorted observations. The RANK procedure can be used to assign ranks to items in a data set in either ascending or descending order. The data do **NOT** have to be sorted before running the RANK procedure. The RANK procedure by itself does not generate printed output. A SAS data table is created when the RANK procedure is run. The RANK procedure does not change the order of the observations in the data table.

ODS Graphics are not available for the RANK procedure.

The syntax of the RANK procedure is as follows:

```
PROC RANK  DATA=tablename  <options>;
VAR  variable-list;                                   list the variables to be ranked
RANKS  new-variable-list;              define the rank variable for each variable ranked
RUN;
```

Options on the PROC RANK statement include the following:

OUT=*newsetname*	creates a data set containing the original data set items and the newly created ranks
DESCENDING	reverses ranking from largest to smallest; SAS will rank in ascending order by default
TIES =	controls how tied observations are to be ranked. Here you must specify MEAN, HIGH, or LOW following the equal sign. For non-parametric tests, average ranks are usually used in the analysis.

There are other options available. See SAS Help and Documentation.

NOTE: There must be the same number of variables in the RANKS statement as there is in the VAR statement. The order of the variables listed in the RANK statement is important. The first variable listed takes ranks of the first variable listed in the VAR statement, and so on.

Objective 4: Rank the response variables X and Y in ascending order. Print the data and the associated ranks. Since no ODS Graphics are a part of the RANK procedure, use the Listing Output.

```
DM 'LOG; CLEAR; ODSRESULTS; CLEAR; ';

DATA  four;
INPUT  X   Y   @@;
DATALINES;
25    41
33    37
27    37
25    29
```

```
42    37
;
PROC  RANK   DATA=four   OUT=new;
VAR  X   Y;
RANKS   RX   RY;
PROC  PRINT   DATA=new;
TITLE 'Objective 4';
RUN;
QUIT;
```

Objective 4

Obs	X	Y	RX	RY
1	25	41	1.5	5
2	33	37	4.0	3
3	27	37	3.0	3
4	25	29	1.5	1
5	42	37	5.0	3

Note how ties are dealt with for each variable. By default, the average rank is used when there are ties. For example, for Y 37 is the 2nd, 3rd, and 4th ordered scores. Therefore, 37 should receive the average of 2, 3, and 4 which is 3. The next ordered score would be in position 5 and should receive rank 5. RX and RY were used as variable names simply so they could be recognized as the rank associated with a given variable. X and Y do not have to be a part of the rank variable names.

Objective 5: Run the above program with each of the following options: TIES=HIGH, TIES=HIGH DESCENDING, TIES=LOW and TIES=LOW DESCENDING options and note the effects of each of the options.

Objective 6: Rank the X observations in ascending order and the Y observations in descending order. Print the data in ascending order according to X along with the associated ranks. Use only one PRINT procedure.

There are two options presented for Objective 6. Either is OK.

Option 1: Track the data sets by name through this program. Note that the data tables must be sorted before the data MERGE.

```
DATA  four;
INPUT  X   Y   @@;
DATALINES;
25    41
33    37
27    37
25    29
42    37
```

```
;
PROC  RANK   DATA=four   OUT=new ;
VAR  X ;
RANKS   RX  ;

PROC RANK DATA=four OUT=new2 DESCENDING;
VAR Y;
RANKS RY;

PROC SORT DATA=new; BY X Y;
PROC SORT DATA=new2; BY X Y ;
DATA combine;
MERGE new new2; BY X Y;

PROC  PRINT   DATA=combine;
TITLE 'Objective 6 - Option 1';
RUN;
QUIT;
```

Option 2: In this option, notice that X, Y, and the ranks for X are in NEW. NEW is then read into a RANK procedure, and the ranks of Y are added to the variables in NEW to form a data set NEW2.

```
DATA  four;
INPUT  X   Y   @@;
DATALINES;
25    41
33    37
27    37
25    29
42    37
;
PROC  RANK   DATA=three   OUT=new ;
VAR  X ;
RANKS   RX  ;

PROC RANK DATA=new OUT=new2 DESCENDING;
VAR Y;
RANKS RY;

PROC  PRINT   DATA=new2;
TITLE 'Objective 6 - Option 2';
RUN;
QUIT;
```

Can you think of a third or fourth option?