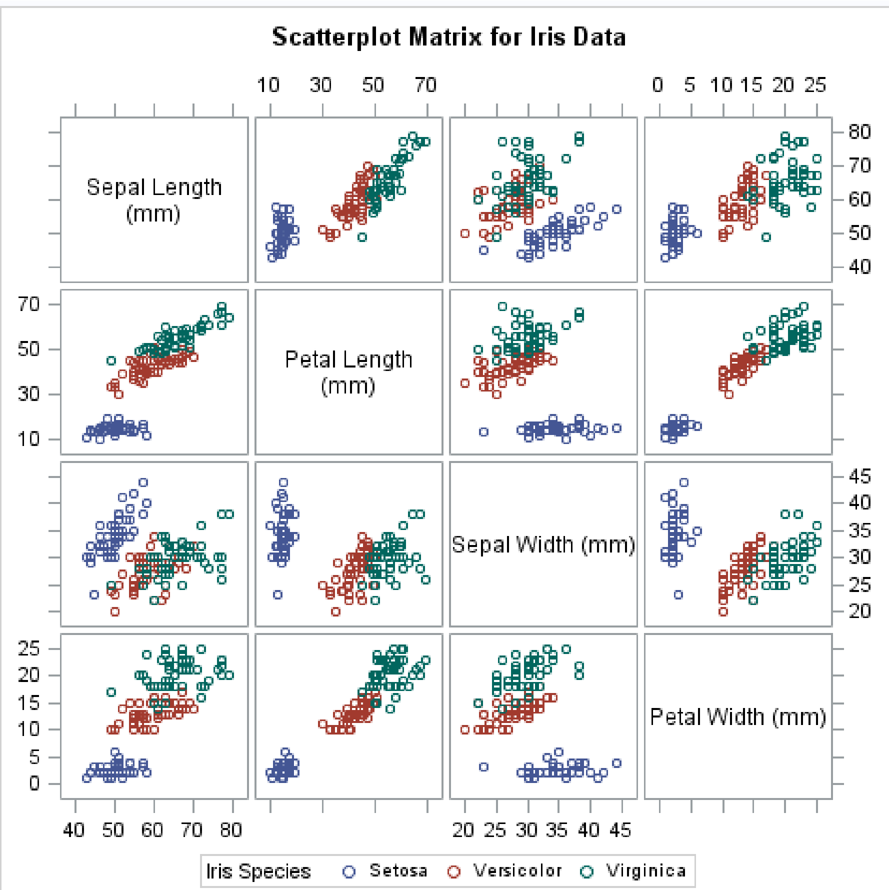
SGSCATTER Procedure
Example 1: Creating a Scatter Plot Matrix

Features:	MATRIX statement GROUP option
Sample library member:	SGSCMAT
Note:	For information about the SAS Sample Library, see About the SASHELP and the SAS Sample Library.

This example shows a scatter plot matrix with grouped data.

Output



Program

```
proc sgscatter data=sashelp.iris;
  title "Scatterplot Matrix for Iris Data";
  matrix sepallength petallength sepalwidth petalwidth
        / group=species;
run;
title;
```

Program Description

Set the title and footnote and create the scatter plot matrix. In the MATRIX statement, the GROUP = option groups the data by the SPECIES variable.

```
proc sgscatter data=sashelp.iris;
  title "Scatterplot Matrix for Iris Data";
  matrix sepallength petallength sepalwidth petalwidth
        / group=species;
run;
title;
```

The UNIVARIATE Procedure

ODS Table Names

PROC UNIVARIATE assigns a name to each table that it creates. You can use these names to reference the table when you use the Output Delivery System (ODS) to select tables and create output data sets.

Table 4.40: ODS Tables Produced with the PROC UNIVARIATE Statement

ODS Table Name	Description	Option
BasicIntervals	confidence intervals for mean, standard deviation, variance	CIBASIC
BasicMeasures	measures of location and variability	default
ExtremeObs	extreme observations	default
ExtremeValues	extreme values	NEXTRVAL=
Frequencies	frequencies	FREQ
LocationCounts	counts used for sign test and signed rank test	LOCCOUNT
MissingValues	missing values	default, if missing values exist
Modes	modes	MODES
Moments	sample moments	default
Plots	line printer plots	PLOTS
Quantiles	quantiles	default
RobustScale	robust measures of scale	ROBUSTSCALE
SSPlots	line printer side-by-side box plots	PLOTS (with BY statement)
TestsForLocation	tests for location	default
TestsForNormality	tests for normality	NORMALTEST
TrimmedMeans	trimmed means	TRIMMED=
WinsorizedMeans	Winsorized means	WINSORIZED=

Table 4.41: ODS Tables Produced with the HISTOGRAM Statement

ODS Table Name	Description	Option
Bins	histogram bins	MIDPERCENTS secondary option
FitQuantiles	quantiles of fitted distribution	any distribution option
GoodnessOfFit	goodness-of-fit tests for fitted distribution	any distribution option
HistogramBins	histogram bins	MIDPERCENTS option
ParameterEstimates	parameter estimates for fitted distribution	any distribution option

Last updated: August 03, 2017

The UNIVARIATE Procedure

Example 4.12 Testing for Location

This example, which is a continuation of Example 4.9, illustrates how to carry out three tests for location: the Student's t test, the sign test, and the Wilcoxon signed rank test. These tests are discussed in the section Tests for Location.

The following statements demonstrate the tests for location by using the *Heights* data set introduced in Example 4.9. Because the data consists of adult female heights, the researchers are not interested in testing whether the mean of the population is equal to zero inches, which is the default μ_0 value. Instead, they are interested in testing whether the mean is equal to 66 inches. The following statements test the null hypothesis $H_0: \mu_0 = 66$:

```
title 'Analysis of Female Height Data';
ods select TestsForLocation LocationCounts;
proc univariate data=Heights mu0=66 loccount;
   var Height;
run;
```

The ODS SELECT statement restricts the output to the "TestsForLocation" and "LocationCounts" tables; see the section ODS Table Names. The MU0= option specifies the null hypothesis value of μ_0 for the tests for location; by default, $\mu_0 = 0$. The LOCCOUNT option produces the table of the number of observations greater than, not equal to, and less than 66 inches.

Output 4.12.1 contains the results of the tests for location. All three tests are highly significant, causing the researchers to reject the hypothesis that the mean is 66 inches.

A sample program for this example, *uniex07.sas*, is available in the SAS Sample Library for Base SAS software.

Output 4.12.1: Tests for Location with MU0=66 and LOCCOUNT

Analysis of Female Height Data

The UNIVARIATE Procedure
Variable: Height (Height (in))

Tests for Location: Mu0=66				
Test	**Statistic**		**p Value**	
Student's t	**t**	-5.67065	**Pr > |t|**	<.0001
Sign	**M**	-20	**Pr >= |M|**	<.0001
Signed Rank	**S**	-849	**Pr >= |S|**	<.0001

Location Counts: Mu0=66.00	
Count	**Value**
Num Obs > Mu0	16
Num Obs ^= Mu0	72
Num Obs < Mu0	56

Version ⌄ Feedback

The UNIVARIATE Procedure

Example 4.7 Saving Summary Statistics in an OUT= Output Data Set

This example illustrates how to save summary statistics in an output data set. The following statements create a data set named *Belts*, which contains the breaking strengths (*Strength*) and widths (*Width*) of a sample of 50 automotive seat belts:

```
data Belts;
   label Strength = 'Breaking Strength (lb/in)'
         Width    = 'Width in Inches';
   input Strength Width @@;
   datalines;
1243.51  3.036  1221.95  2.995  1131.67  2.983  1129.70  3.019
1198.08  3.106  1273.31  2.947  1250.24  3.018  1225.47  2.980
1126.78  2.965  1174.62  3.033  1250.79  2.941  1216.75  3.037
1285.30  2.893  1214.14  3.035  1270.24  2.957  1249.55  2.958
1166.02  3.067  1278.85  3.037  1280.74  2.984  1201.96  3.002
1101.73  2.961  1165.79  3.075  1186.19  3.058  1124.46  2.929
1213.62  2.984  1213.93  3.029  1289.59  2.956  1208.27  3.029
1247.48  3.027  1284.34  3.073  1209.09  3.004  1146.78  3.061
1224.03  2.915  1200.43  2.974  1183.42  3.033  1195.66  2.995
1258.31  2.958  1136.05  3.022  1177.44  3.090  1246.13  3.022
1183.67  3.045  1206.50  3.024  1195.69  3.005  1223.49  2.971
1147.47  2.944  1171.76  3.005  1207.28  3.065  1131.33  2.984
1215.92  3.003  1202.17  3.058
;
```

The following statements produce two output data sets containing summary statistics:

```
proc univariate data=Belts noprint;
   var Strength Width;
   output out=Means         mean=StrengthMean WidthMean;
   output out=StrengthStats mean=StrengthMean std=StrengthSD
                            min=StrengthMin   max=StrengthMax;
run;
```

When you specify an OUTPUT statement, you must also specify a VAR statement. You can use multiple OUTPUT statements with a single procedure statement. Each OUTPUT statement creates a new data set with the name specified by the OUT= option. In this example, two data sets, *Means* and *StrengthStats*, are created. See Output 4.7.1 for a listing of *Means* and Output 4.7.2 for a listing of *StrengthStats*.

Output 4.7.1: Listing of Output Data Set Means

Analysis of Speeding Data

Obs	StrengthMean	WidthMean
1	1205.75	3.00584

Output 4.7.2: Listing of Output Data Set StrengthStats

Analysis of Speeding Data

Obs	StrengthMean	StrengthSD	StrengthMax	StrengthMin
1	1205.75	48.3290	1289.59	1101.73

Summary statistics are saved in an output data set by specifying *keyword=names* after the OUT= option. In the preceding statements, the first OUTPUT statement specifies the *keyword* MEAN followed by the *names StrengthMean* and *WidthMean*. The second OUTPUT statement specifies the *keywords* MEAN, STD, MAX, and MIN, for which the *names StrengthMean*, *StrengthSD*, *StrengthMax*, and *StrengthMin* are given.

The *keyword* specifies the statistic to be saved in the output data set, and the *names* determine the names for the new variables. The first *name* listed after a keyword contains that statistic for the first variable listed in the VAR statement; the second *name* contains that statistic for the second variable in the VAR statement, and so on.

The data set *Means* contains the mean of *Strength* in a variable named *StrengthMean* and the mean of *Width* in a variable named *WidthMean*. The data set *StrengthStats* contains the mean, standard deviation, maximum value, and minimum value of *Strength* in the variables *StrengthMean*, *StrengthSD*, *StrengthMax*, and *StrengthMin*, respectively.

See the section OUT= Output Data Set in the OUTPUT Statement for more information about OUT= output data sets.

A sample program for this example, *uniex06.sas*, is available in the SAS Sample Library for Base SAS software.

The TTEST Procedure

ODS Table Names

PROC TTEST assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 122.9. For more information about ODS, see Chapter 20: Using the Output Delivery System.

Table 122.9: ODS Tables Produced by PROC TTEST

ODS Table Name	Description	Syntax
ConfLimits	100(1 – α)% confidence limits for means, standard deviations, and/or coefficients of variation	By default
Equality	Tests for equality of variance	CLASS statement or VAR / CROSSOVER=
EquivLimits	100(1 – 2 α)% confidence limits for means	PROC TTEST TOST
EquivTests	Equivalence t tests	PROC TTEST TOST
Statistics	Univariate summary statistics	By default
TTests	t tests	By default
Bootstrap	Bootstrap standard error and bias estimates and 100(1 – α)% bootstrap confidence limits	BOOTSTRAP statement

The TTEST Procedure

Example 122.1 Using Summary Statistics to Compare Group Means

(View the complete code for this example.)

This example, taken from Huntsberger and Billingsley (1989), compares two grazing methods using 32 steers. Half of the steers are allowed to graze continuously while the other half are subjected to controlled grazing time. The researchers want to know if these two grazing methods affect weight gain differently. The data are read by the following DATA step:

```
data graze;
   length GrazeType $ 10;
   input GrazeType $ WtGain @@;
   datalines;
controlled  45   controlled  62
controlled  96   controlled 128
controlled 120   controlled  99
controlled  28   controlled  50
controlled 109   controlled 115
controlled  39   controlled  96
controlled  87   controlled 100
controlled  76   controlled  80
continuous  94   continuous  12
continuous  26   continuous  89
continuous  88   continuous  96
continuous  85   continuous 130
continuous  75   continuous  54
continuous 112   continuous  69
continuous 104   continuous  95
continuous  53   continuous  21
;
```

The variable *GrazeType* denotes the grazing method: "controlled" is controlled grazing and "continuous" is continuous grazing. The dollar sign ($) following *GrazeType* makes it a character variable, and the trailing at signs (@@) tell the procedure that there is more than one observation per line.

If you have summary data—that is, just means and standard deviations, as computed by PROC MEANS—then you can still use PROC TTEST to perform a simple t test analysis. This example demonstrates this mode of input for PROC TTEST. Note, however, that graphics are unavailable when summary statistics are used as input.

The MEANS procedure is invoked to create a data set of summary statistics with the following statements:

```
proc sort;
   by GrazeType;
run;

proc means data=graze noprint;
   var WtGain;
   by GrazeType;
   output out=newgraze;
run;
```

The NOPRINT option eliminates all printed output from the MEANS procedure. The VAR statement tells PROC MEANS to compute summary statistics for the *WtGain* variable, and the BY statement requests a separate set of summary statistics for each level of *GrazeType*. The OUTPUT OUT= statement tells PROC MEANS to put the summary statistics into a data set called *newgraze* so that it can be used in subsequent procedures. This new data set is displayed in Output 122.1.1 by using PROC PRINT as follows:

```
proc print data=newgraze;
run;
```

The *_STAT_* variable contains the names of the statistics, and the *GrazeType* variable indicates which group the statistic is from.

Output 122.1.1: Output Data Set of Summary Statistics

Obs	GrazeType	_TYPE_	_FREQ_	_STAT_	WtGain
1	continuous	0	16	N	16.000
2	continuous	0	16	MIN	12.000
3	continuous	0	16	MAX	130.000
4	continuous	0	16	MEAN	75.188
5	continuous	0	16	STD	33.812
6	controlled	0	16	N	16.000
7	controlled	0	16	MIN	28.000
8	controlled	0	16	MAX	128.000
9	controlled	0	16	MEAN	83.125
10	controlled	0	16	STD	30.535

The following statements invoke PROC TTEST with the *newgraze* data set, as denoted by the DATA= option:

```
proc ttest data=newgraze;
   class GrazeType;
   var WtGain;
run;
```

The CLASS statement contains the variable that distinguishes between the groups being compared, in this case *GrazeType*. The summary statistics and confidence intervals are displayed first, as shown in Output 122.1.2.

Output 122.1.2: Summary Statistics and Confidence Limits

The TTEST Procedure

Variable: WtGain

GrazeType	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
continuous		16	75.1875	33.8117	8.4529	12.0000	130.0
controlled		16	83.1250	30.5350	7.6337	28.0000	128.0
Diff (1-2)	**Pooled**		-7.9375	32.2150	11.3897		
Diff (1-2)	**Satterthwaite**		-7.9375		11.3897		

GrazeType	Method	Mean	95% CL Mean	Std Dev	95% CL Std Dev

GrazeType	Method	Mean	95% CL Mean		Std Dev	95% CL Std Dev	
continuous		75.1875	57.1705	93.2045	33.8117	24.9768	52.3300
controlled		83.1250	66.8541	99.3959	30.5350	22.5563	47.2587
Diff (1-2)	Pooled	-7.9375	-31.1984	15.3234	32.2150	25.7434	43.0609
Diff (1-2)	Satterthwaite	-7.9375	-31.2085	15.3335			

In Output 122.1.2, The GrazeType column specifies the group for which the statistics are computed. For each class, the sample size, mean, standard deviation and standard error, and maximum and minimum values are displayed. The confidence bounds for the mean are also displayed.

Output 122.1.3 shows the results of tests for equal group means and equal variances.

Output 122.1.3: t Tests

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	30	-0.70	0.4912
Satterthwaite	Unequal	29.694	-0.70	0.4913

Equality of Variances				
Method	Num DF	Den DF	F Value	Pr > F
Folded F	15	15	1.23	0.6981

A group test statistic for the equality of means is reported for both equal and unequal variances. Both tests indicate a lack of evidence for a significant difference between grazing methods (t = –0.70 and p = 0.4912 for the pooled test, t = –0.70 and p = 0.4913 for the Satterthwaite test). The equality of variances test does not indicate a significant difference in the two variances $(F' = 1.23, p = 0.6981)$. Note that this test assumes that the observations in both data sets are normally distributed; this assumption can be checked in PROC UNIVARIATE by using the NORMAL option with the raw data.

Although the ability to use summary statistics as input is useful if you lack access to the original data, some of the output that would otherwise be produced in an analysis on the original data is unavailable. There are also limitations on the designs and distributional assumptions that can be used with summary statistics as input. For more information, see the section Input Data Set of Statistics.

Version ⌄ Feedback

The CORR Procedure

Example 2.1 Computing Four Measures of Association

This example produces a correlation analysis with descriptive statistics and four measures of association: the Pearson product-moment correlation, the Spearman rank-order correlation, Kendall's tau-b coefficients, and Hoeffding's measure of dependence, D.

The *Fitness* data set created in the section Getting Started: CORR Procedure contains measurements from a study of physical fitness of 31 participants. The following statements request all four measures of association for the variables *Weight*, *Oxygen*, and *Runtime*:

```
ods graphics on;
title 'Measures of Association for a Physical Fitness Study';
proc corr data=Fitness pearson spearman kendall hoeffding
          plots=matrix(histogram);
   var Weight Oxygen RunTime;
run;
```

Note that Pearson correlations are computed by default only if all three nonparametric correlations (SPEARMAN, KENDALL, and HOEFFDING) are not specified. Otherwise, you need to specify the PEARSON option explicitly to compute Pearson correlations.

The "Simple Statistics" table in Output 2.1.1 displays univariate descriptive statistics for analysis variables. By default, observations with nonmissing values for each variable are used to derive the univariate statistics for that variable. When nonparametric measures of association are specified, the procedure displays the median instead of the sum as an additional descriptive measure.

Output 2.1.1: Simple Statistics

Measures of Association for a Physical Fitness Study

The CORR Procedure

3 Variables:	Weight Oxygen RunTime

Simple Statistics

Variable	N	Mean	Std Dev	Median	Minimum	Maximum
Weight	31	77.44452	8.32857	77.45000	59.08000	91.63000
Oxygen	29	47.22721	5.47718	46.67200	37.38800	60.05500
RunTime	29	10.67414	1.39194	10.50000	8.17000	14.03000

The "Pearson Correlation Coefficients" table in Output 2.1.2 displays Pearson correlation statistics for pairs of analysis variables. The Pearson correlation is a parametric measure of association for two continuous random variables. When there are missing data, the number of observations used to calculate the correlation can vary.

Output 2.1.2: Pearson Correlation Coefficients

Pearson Correlation Coefficients
Prob > |r| under H0: Rho=0
Number of Observations

Pearson Correlation Coefficients Prob > |r| under H0: Rho=0 Number of Observations			
	Weight	Oxygen	RunTime
Weight	1.00000 31	-0.15358 0.4264 29	0.20072 0.2965 29
Oxygen	-0.15358 0.4264 29	1.00000 29	-0.86843 <.0001 28
RunTime	0.20072 0.2965 29	-0.86843 <.0001 28	1.00000 29

The table shows that the Pearson correlation between *Runtime* and *Oxygen* is –0.86843, which is significant with a p-value less than 0.0001. This indicates a strong negative linear relationship between these two variables. As *Runtime* increases, *Oxygen* decreases linearly.

The Spearman rank-order correlation is a nonparametric measure of association based on the ranks of the data values. The "Spearman Correlation Coefficients" table in Output 2.1.3 displays results similar to those of the "Pearson Correlation Coefficients" table in Output 2.1.2.

Output 2.1.3: Spearman Correlation Coefficients

Spearman Correlation Coefficients Prob > |r| under H0: Rho=0 Number of Observations			
	Weight	Oxygen	RunTime
Weight	1.00000 31	-0.06824 0.7250 29	0.13749 0.4769 29
Oxygen	-0.06824 0.7250 29	1.00000 29	-0.80131 <.0001 28
RunTime	0.13749 0.4769 29	-0.80131 <.0001 28	1.00000 29

Kendall's tau-b is a nonparametric measure of association based on the number of concordances and discordances in paired observations. The "Kendall Tau b Correlation Coefficients" table in Output 2.1.4 displays results similar to those of the "Pearson Correlation Coefficients" table in Output 2.1.2.

Output 2.1.4: Kendall's Tau-b Correlation Coefficients

Kendall Tau b Correlation Coefficients Prob > |tau| under H0: Tau=0 Number of Observations			
	Weight	Oxygen	RunTime
Weight	1.00000 31	-0.00988 0.9402 29	0.06675 0.6123 29
Oxygen	-0.00988 0.9402 29	1.00000 29	-0.62434 <.0001 28

Kendall Tau b Correlation Coefficients Prob > |tau| under H0: Tau=0 Number of Observations			
	Weight	**Oxygen**	**RunTime**
RunTime	0.06675 0.6123 29	-0.62434 <.0001 28	1.00000 29

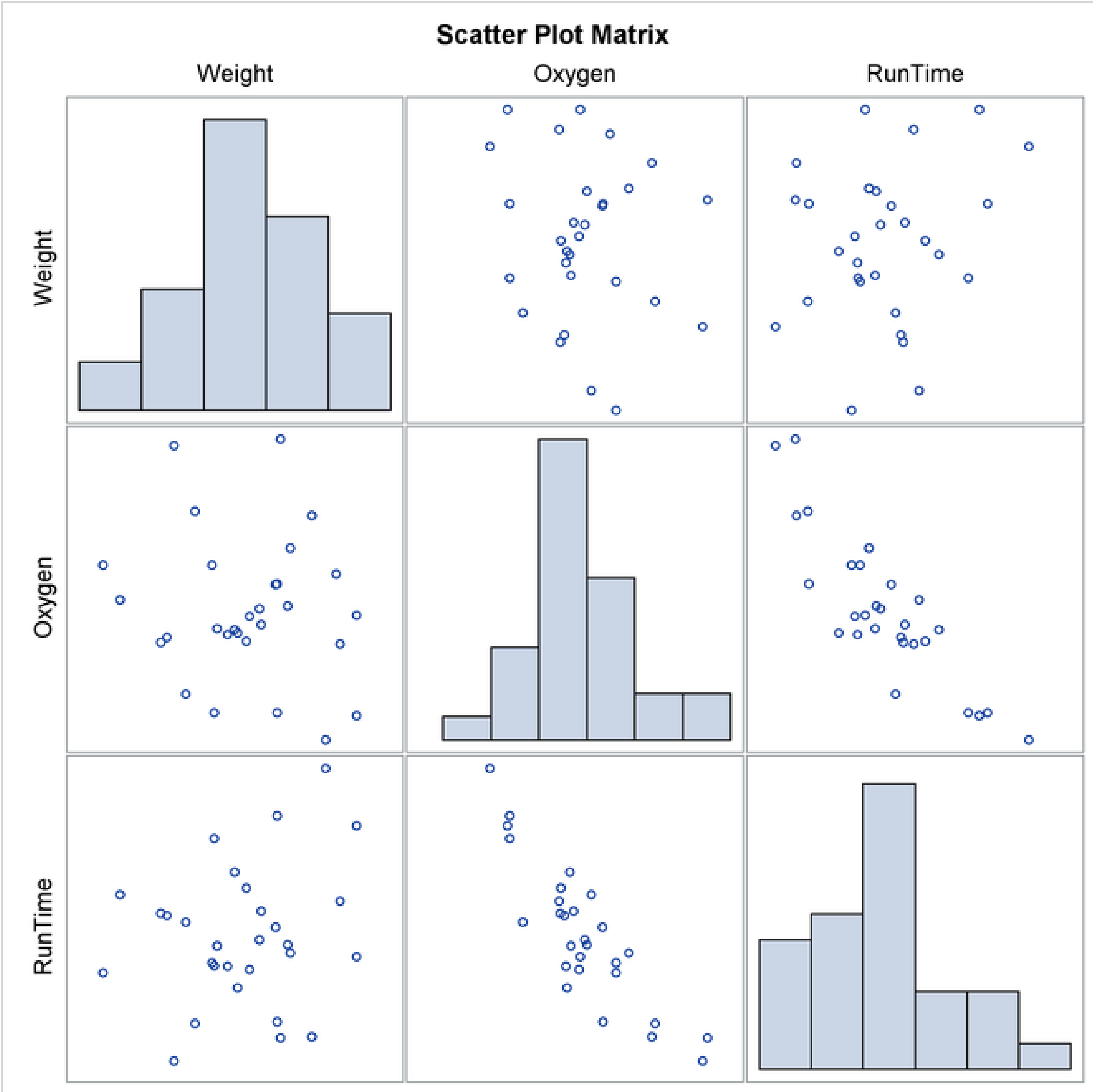
Hoeffding's measure of dependence, D, is a nonparametric measure of association that detects more general departures from independence. Without ties in the variables, the values of the D statistic can vary between –0.5 and 1, with 1 indicating complete dependence. Otherwise, the D statistic can result in a smaller value. The "Hoeffding Dependence Coefficients" table in Output 2.1.5 displays Hoeffding dependence coefficients. Since ties occur in the variable *Weight*, the D statistic for the *Weight* variable is less than 1.

Output 2.1.5: Hoeffding's Dependence Coefficients

Hoeffding Dependence Coefficients Prob > D under H0: D=0 Number of Observations			
	Weight	**Oxygen**	**RunTime**
Weight	0.97690 <.0001 31	-0.00497 0.5101 29	-0.02355 1.0000 29
Oxygen	-0.00497 0.5101 29	1.00000 29	0.23449 <.0001 28
RunTime	-0.02355 1.0000 29	0.23449 <.0001 28	1.00000 29

When you use the PLOTS=MATRIX(HISTOGRAM) option, the CORR procedure displays a symmetric matrix plot for the analysis variables listed in the VAR statement (Output 2.1.6).

Output 2.1.6: Symmetric Scatter Plot Matrix



The strong negative linear relationship between *Oxygen* and *Runtime* is evident in Output 2.1.6.

Note that this graphical display is requested by enabling ODS Graphics and by specifying the PLOTS= option. For more information about ODS Graphics, see Chapter 21: Statistical Graphics Using ODS in *SAS/STAT 14.3 User's Guide*.

Last updated: August 03, 2017

The CORR Procedure

Example 2.7 Saving Correlations in an Output Data Set

The following statements compute Pearson correlations:

```
title 'Correlations for a Fitness and Exercise Study';
proc corr data=Fitness nomiss outp=CorrOutp;
   var weight oxygen runtime;
run;
```

The NOMISS option excludes observations with missing values of the VAR statement variables from the analysis—that is, the same set of 28 observations is used to compute the correlation for each pair of variables. The OUTP= option creates an output data set named *CorrOutp* that contains the Pearson correlation statistics.

The "Pearson Correlation Coefficients" table in Output 2.7.1 displays the correlation and the p-value under the null hypothesis of zero correlation.

Output 2.7.1: Pearson Correlation Coefficients

Correlations for a Fitness and Exercise Study

The CORR Procedure

Pearson Correlation Coefficients, N = 28 Prob > |r| under H0: Rho=0			
	Weight	**Oxygen**	**RunTime**
Weight	1.00000	-0.18419 0.3481	0.19505 0.3199
Oxygen	-0.18419 0.3481	1.00000	-0.86843 <.0001
RunTime	0.19505 0.3199	-0.86843 <.0001	1.00000

The following statements display (in Output 2.7.2) the output data set:

```
title 'Output Data Set from PROC CORR';
proc print data=CorrOutp noobs;
run;
```

Output 2.7.2: OUTP= Data Set with Pearson Correlations

Output Data Set from PROC CORR

TYPE	_NAME_	Weight	Oxygen	RunTime
MEAN		77.2168	47.1327	10.6954
STD		8.4495	5.5535	1.4127
N		28.0000	28.0000	28.0000

TYPE	_NAME_	Weight	Oxygen	RunTime
CORR	Weight	1.0000	-0.1842	0.1950
CORR	Oxygen	-0.1842	1.0000	-0.8684
CORR	RunTime	0.1950	-0.8684	1.0000

The output data set has the default type CORR and can be used as an input data set for regression or other statistical procedures. For example, the following statements request a regression analysis using *CorrOutp*, without reading the original data in the REG procedure:

```
title 'Input Type CORR Data Set from PROC REG';
proc reg data=CorrOutp;
   model runtime= weight oxygen;
run;
```

The following statements generate the same results as the preceding statements:

```
proc reg data=Fitness;
   model runtime= weight oxygen;
run;
```


Last updated: August 03, 2017

Wilcoxon Rank Sum Test Explanation

SAS Code Example:

```
proc npar1way data=dataName wilcoxon;
class classVariable;
var variable1;
ods select WilcoxonScores WilcoxonTest;
run;
```

The **null hypothesis** is that H0: there is no difference between the two populations (or distributions). Another way to think of this is H0: the two populations have similar values.

The **two-sided alternative** is that H2: there is a difference between the two populations (or distributions). Another way to think of this is H2: the two populations do not have similar values. If the p-value is less than 0.05, then we reject the null in favor of this two sided alternative.

Look at the value of the Statistic (W) in the table "Wilcoxon Two-Sample Test". This statistic will match one of the values for the “Sum of Scores” column of the table "Wilcoxon Scores (Rank Sums)." This statistic will tell you which distribution (or population) we are focusing on. Look at the mean value (muW) for that distribution which can be found in the column "Expected Value under H0" of the table "Wilcoxon Scores (Rank Sums)." Look at the standard deviation for that distribution which can be found in the column “Std Dev under H0.” The Z value is Z=W-muW/sigmaW.

SAS will show us a different **one-sided alternative** depending on the sign of the Z value. Look at the value of Z in the "Wilcoxon Two-Sample Test" table.

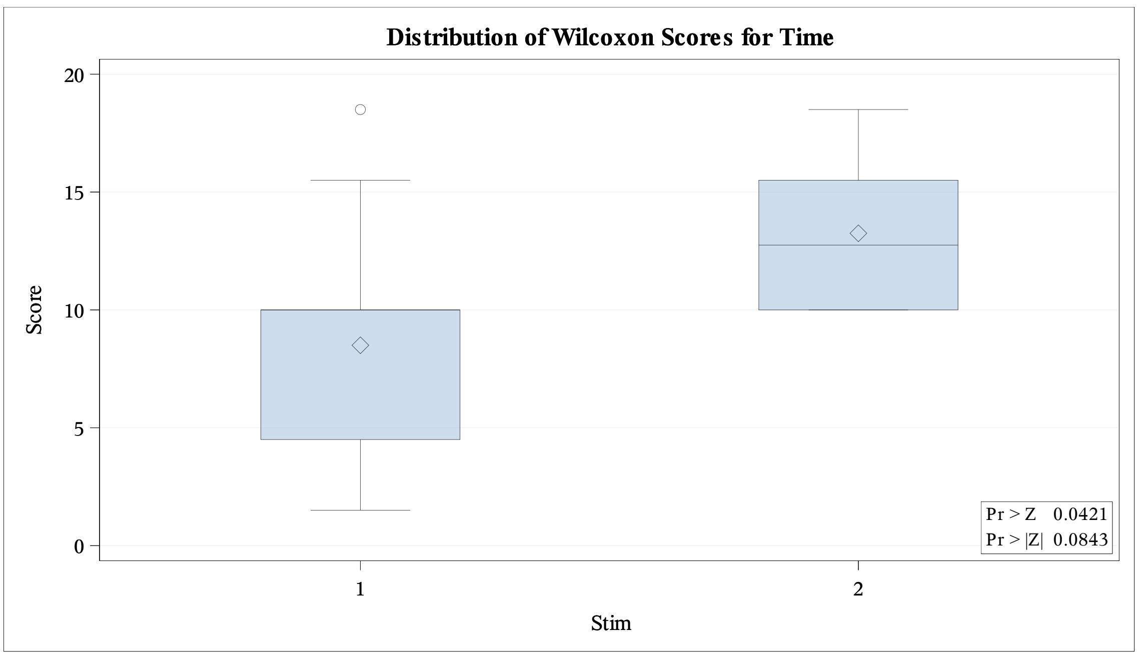
If Z is positive, then that means SAS will show you that the alternative hypothesis is Pr > Z. If that p-value is less than 0.05, then that means the population we focus on has much larger values than the other population. If Pr > Z is greater than 0.05, then we do not reject the null hypothesis.

Wilcoxon Scores (Rank Sums) for Variable Time Classified by Variable Stim

Stim	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
1	13	110.50	130.0	11.004784	8.500
2	6	79.50	60.0	11.004784	13.250

Average scores were used for ties.

Wilcoxon Two-Sample Test	
Statistic	79.5000
Normal Approximation	
Z	1.7265
One-Sided Pr > Z	0.0421
Two-Sided Pr > |Z|	0.0843
t Approximation	
One-Sided Pr > Z	0.0507
Two-Sided Pr > |Z|	0.1014
Z includes a continuity correction of 0.5.	

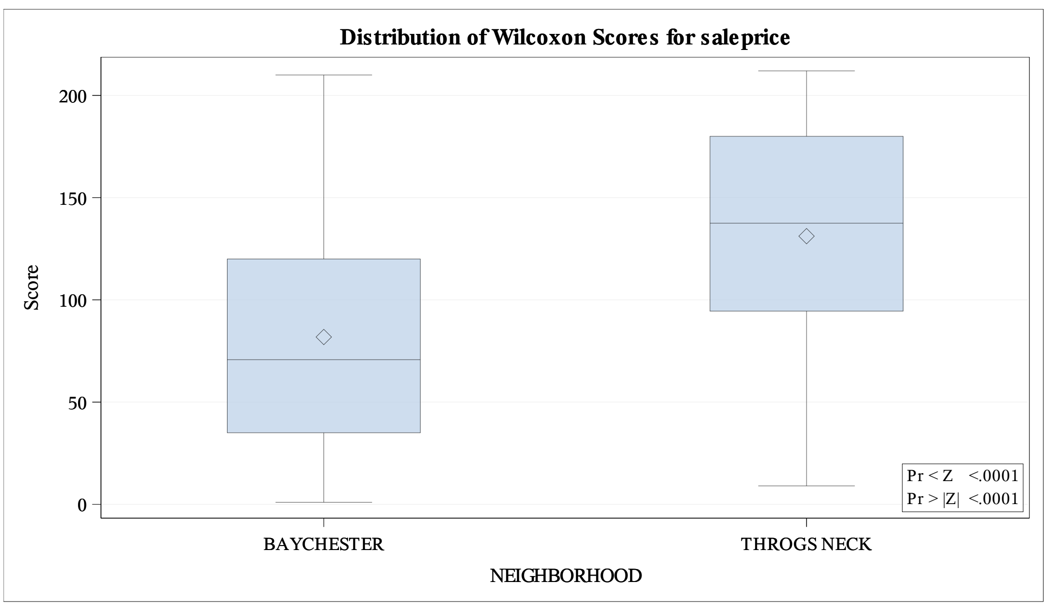


For the above example, the population that we focus on is Stim 2 since the statistic that matches Stim 2's Sum of Scores is 79.5 and Z is positive. The two-sided p-value is 0.08 so in this case we would not reject the null. Hence there is not significance evidence that Stim 1 and 2 have different values. The one-sided (Pr > Z) p-value is 0.04 so in this case we would reject the null in favor of the alternative that Stim 2 has larger values than Stim 1. The box plot tells us the same thing.

If Z is negative, then that means SAS will show you that the alternative hypothesis is Pr < Z. If that p-value is less than 0.05, then that means the population we focus on has much smaller values than the other population. If Pr < Z is greater than 0.05, then we do not reject the null hypothesis.

Wilcoxon Scores (Rank Sums) for Variable saleprice Classified by Variable NEIGHBORHOOD					
NEIGHBORHOOD	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
BAYCHESTER	106	8674.50	11289.0	446.550221	81.834906
THROGS NECK	106	13903.50	11289.0	446.550221	131.165094
Average scores were used for ties.					

Wilcoxon Two-Sample Test	
Statistic	8674.5000
Normal Approximation	
Z	-5.8538
One-Sided Pr < Z	<.0001
Two-Sided Pr > |Z|	<.0001
t Approximation	
One-Sided Pr < Z	<.0001
Two-Sided Pr > |Z|	<.0001
Z includes a continuity correction of 0.5.	



For the above example, the population that we focus on is Baychester since the statistic is 8674.5 and Z is negative. The two-sided p-value is <0.001 so in this case we would reject the null. Hence there is significance evidence that Baychester and Throgs Neck have different values. The one-sided (Pr < Z) p-value is <0.001 so in this case we would reject the null in favor of the alternative that Baychester has smaller values than Throgs Neck. The box plot tells us the same thing.