IBM SPSS Statistics V27 - Beta Documentation

Note

Before using this information and the product it supports, read the information in Chapter 4, “Notices,” on page 183.

Product Information

This edition applies to version V27, release 0, modification 0 of IBM® SPSS® Statistics and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation .
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. Overview.......... 1
What's new in version 27.......... 1

Chapter 2. Analyze procedures.......... 7
Bivariate Correlations.......... 7
Bivariate Correlations Options.......... 8
CORRELATIONS and NONPAR CORR Command Additional Features.......... 8
Crosstabs.......... 8
Crosstabs layers.......... 9
Crosstabs clustered bar charts.......... 9
Crosstabs displaying layer variables in table layers.......... 9
Crosstabs statistics.......... 10
Crosstabs cell display.......... 11
Crosstabs table format.......... 12
Frequencies.......... 12
Frequencies Statistics.......... 13
Frequencies Charts.......... 14
Frequencies Format.......... 14
Power Analysis.......... 14
Means.......... 16
Proportions.......... 26
Correlations.......... 34
Regression.......... 41

Chapter 3. Command enhancements.......... 45
CORRELATIONS 45
Overview 45
Example 46
VARIABLES Subcommand 46
PRINT Subcommand 46
STATISTICS Subcommand 47
MISSING Subcommand 47
MATRIX Subcommand 47
MATRIX-END MATRIX 49
Overview 49
Terminology 50
Matrix Variables 50
Syntax of Matrix Language 51
Matrix Operations 52
MATRIX and Other Commands 55
MATRIX and END MATRIX Commands 56
COMPUTE Statement 56
CALL Statement 67
PRINT Statement 67
Matrix Control Structures 70
READ Statement: Reading Character Data 72
WRITE Statement: Writing Character Data 74
GET Statement: Reading IBM SPSS Statistics Data Files 76
SAVE Statement: Writing IBM SPSS Statistics Data Files 78
MGET Statement: Reading Matrix Data Files 80

- MSAVE Statement: Writing Matrix Data Files 81
- DISPLAY Statement 84
- RELEASE Statement 84
- Macros Using the Matrix Language 84

NONPAR CORR 84
- Overview 85
- Examples 85
- VARIABLES Subcommand 86
- PRINT Subcommand 86
- SAMPLE Subcommand 87
- MISSING Subcommand 87
- MATRIX Subcommand 87

NPTESTS 89
- Overview 90
- MISSING Subcommand 91
- CRITERIA Subcommand 91
- ONESAMPLE Subcommand 92
- INDEPENDENT Subcommand 95
- RELATED Subcommand 97

NPAR TESTS 100
- Overview 101
- BINOMIAL Subcommand 102
- CHISQUARE Subcommand 103
- COCHRAN Subcommand 104
- FRIEDMAN Subcommand 104
- J-T Subcommand 105
- K-S Subcommand (One-Sample) 105
- K-S Subcommand (Two-Sample) 106
- KS_SIM Subcommand (One-Sample) 107
- K-W Subcommand 107
- KENDALL Subcommand 108
- M-W Subcommand 108
- MCNEMAR Subcommand 109
- MEDIAN Subcommand 109
- MH Subcommand 110
- MOSES Subcommand 111
- RUNS Subcommand 111
- SIGN Subcommand 112
- W-W Subcommand 113
- WILCOXON Subcommand 113
- STATISTICS Subcommand 114
- MISSING Subcommand 114
- SAMPLE Subcommand 114
- METHOD Subcommand 114
- References 115

ONEWAY 115
- Overview 116
- Analysis List 117
- POLYNOMIAL Subcommand 117
- CONTRAST Subcommand 118
- POSTHOC Subcommand 118
- RANGES Subcommand 120
- PLOT MEANS Subcommand 120
- STATISTICS Subcommand 120
- MISSING Subcommand 120
- MATRIX Subcommand 121
- TEMPLATE Subcommand 122
- CRITERIA Subcommand 123

ES Subcommand 123
References 123
OUTPUT MODIFY 123
Overview 125
Basic Operation 126
NAME Keyword 126
SELECT Subcommand 126
IF Subcommand 127
DELETEOBJECT Subcommand 129
INDEXING Subcommand 130
OBJECTPROPERTIES Subcommand 131
TABLE Subcommand 132
TABLECELLS Subcommand 135
GRAPHS Subcommand 139
TEXTS Subcommand 139
REPORT Subcommand 140
POWER ONEWAY ANOVA 141
Overview 141
Example 142
PARAMETERS Subcommand 142
CONTRAST Subcommand 143
PLOT Subcommand 143
POWER MEANS INDEPENDENT 144
Overview 144
Example 145
PARAMETERS Subcommand 145
PLOT Subcommand 146
POWER MEANS ONESAMPLE 146
Overview 147
Example 147
PARAMETERS Subcommand 148
PLOT Subcommand 148
POWER MEANS RELATED 149
Overview 149
Example 150
PARAMETERS Subcommand 150
PLOT Subcommand 151
POWER PARTIALCORR 152
Overview 152
Example 153
PARAMETERS Subcommand 153
PLOT Subcommand 153
POWER PEARSON ONESAMPLE 154
Overview 155
Example 155
PARAMETERS Subcommand 156
PLOT Subcommand 156
POWER PROPORTIONS INDEPENDENT 157
Overview 158
Example 158
PARAMETERS Subcommand 158
PLOT Subcommand 159
POWER PROPORTIONS ONESAMPLE 160
Overview 161
Example 161
PARAMETERS Subcommand 161
PLOT Subcommand 162
POWER PROPORTIONS RELATED 163

Overview 163
Example 164
PARAMETERS Subcommand 164
PLOT Subcommand 165
POWER SPEARMAN ONESAMPLE...166
Overview 166
Example 167
PARAMETERS Subcommand 167
PLOT Subcommand 168
POWER UNIVARIATE LINEAR...169
Overview 169
Example 170
PARAMETERS Subcommand 170
PLOT Subcommand 171
QUANTILE REGRESSION...171
Overview 172
CRITERIA Subcommand 173
MODEL Subcommand 174
SAVE Subcommand 174
PREDICT_EFFECTS Subcommand 174
PREDICTION_TABLES Subcommand 174
PREDICTION_LINES Subcommand 174
PRINT Subcommand 175
PLOT Subcommand 175
OUTFILE Subcommand...175
EXTERNAL Subcommand 175
T-TEST...175
Overview 176
Examples 177
VARIABLES Subcommand 177
TESTVAL Subcommand 177
GROUPS Subcommand 178
PAIRS Subcommand 178
CRITERIA Subcommand 178
ES Subcommand 178
MISSING Subcommand 179
WEIGHTED KAPPA...179
Overview (WEIGHTED KAPPA command)..180
varlist1 BY varlist2 180
CRITERIA Subcommand 180
PRINT Subcommand 181
MISSING Subcommand 181
Trademarks..184

Notices...185
Trademarks..186

Index..187

Chapter 1. Overview

What's new in version 27

Packaging

The Bootstrapping and Data Preparation features are now included in the IBM SPSS Statistics Base edition (Bootstrapping was previously included in the Premium Edition; Data Preparation was previously included in the Professional Edition).

Automatic recovery

Automatic recovery is designed to recover unsaved files and content in instances where the application quits unexpectedly. You can select to enable/disable the automatic recovery feature (the feature is enabled by default), select a time interval (in minutes) between saving files, and view or change the auto-recovery file location.

Upon relaunching SPSS Statistics after an unexpected exit, you are presented with an IBM SPSS Statistics error report, which allows you to enter information about your session prior to the unexpected exit. After leaving the exit report, you are presented with the Document Recovery dialog, which provides options for recovering prior session data or deleting the saved session data.

Issue reporter

The Help menu now provides a **Report Issue** link that launches the IBM SPSS Statistics Issue Reporter dialog. The dialog allows you to enter information regarding any issues you may encounter when using the product. The information you enter is sent to IBM for use in improving the product.

Analyze procedures

Bivariate Correlations

The **Show only the lower triangle** setting was added to main dialog. When the setting is enabled only the correlation matrix table's lower triangle is presented in the output. When not selected, the full correlation matrix table is presented in the output. The setting was introduced to allow table output to adhere to APA style guidelines.

Crosstabs

The **Create APA style table** settings was added to Cell Display dialog. The setting produces a table that adhere to APA style guidelines.

Frequencies

The **Create APA style tables** settings was added to main dialog. The setting produces tables that adhere to APA style guidelines.

Power Analysis

Power analysis plays a pivotal role in a study plan, design, and conduction. The calculation of power is usually before any sample data have been collected, except possibly from a small pilot study. The precise estimation of the power may tell investigators how likely it is that a statistically significant difference will be detected based on a finite sample size under a true alternative hypothesis. If the power is too low, there is little chance of detecting a significant difference, and non-significant results are likely even if real differences truly exist. The new procedures are grouped as follows.

Means

One-Sample T-Test

In one-sample analysis, the observed data are collected as a single random sample. It is assumed that the sample data independently and identically follow a normal distribution with a fixed mean and variance, and draws statistical inference about the mean parameter.

Independent-Samples T-Test

In independent-samples analysis, the observed data contain two independent samples. It is assumed that the data in each sample independently and identically follow a normal distribution with a fixed mean and variance, and draws statistical inference about the difference of the two means.

Paired-Samples T-Test

In paired-samples analysis, the observed data contain two paired and correlated samples, and each case has two measurements. It is assumed that the data in each sample independently and identically follow a normal distribution with a fixed mean and variance, and draws statistical inference about the difference of the two means.

One-Way ANOVA

Analysis of variance (ANOVA) is a statistical method of estimating the means of several populations which are often assumed to be normally distributed. The One-way ANOVA, a common type of ANOVA, is an extension of the two-sample *t*-test. The procedure provides approaches for estimating the power for two types of hypothesis to compare the multiple group means, the overall test, and the test with specified contrasts. The over test focuses on the null hypothesis that all group means are equal. The test with specified contrasts breaks down the overall ANOVA hypotheses into smaller but more describable and useful pieces of the means.

Proportions

One-Sample Binomial Test

The binomial distribution is based on a sequence of Bernoulli trials. It can be used to model those experiments including a fixed number of total trials that are assumed to be independent of each other. Each trial leads to a dichotomous result, with the same probability for a "successful" outcome.

The one-sample binomial test makes statistical inference about the proportion parameter by comparing it with a hypothesized value. The methods for estimating the power for such a test are either the normal approximation or the binomial enumeration.

Related-Samples Binomial Test

The binomial distribution is based on a sequence of Bernoulli trials. It can be used to model those experiments including a fixed number of total trials that are assumed to be independent of each other. Each trial leads to a dichotomous result, with the same probability for a "successful" outcome.

The related-sample binomial estimates the power of McNemar's test to compare two proportion parameters based on the matched pair subjects sampled from two related binomial populations.

Independent-Samples Binomial Test

The binomial distribution is based on a sequence of Bernoulli trials. It can be used to model those experiments including a fixed number of total trials that are assumed to be independent of each other. Each trial leads to a dichotomous result, with the same probability for a "successful" outcome.

The independent-sample binomial test compares two independent proportion parameters.

Correlations

Pearson Product-Moment

Pearson's product-moment correlation coefficient measures the strength of linear association between two scale random variables that are assumed to follow a bivariate normal distribution. By convention, it is a dimensionless quantity and obtained by standardizing the covariance between two continuous variables, thereby ranging between -1 and 1.

The test uses Fisher's asymptotic method to estimate the power for the one-sample Pearson correlation.

Spearman Rank-Order

Spearman rank-order correlation coefficient is a rank-based nonparametric statistic to measure the monotonic relationship between two variables that are usually censored and not normally distributed. The Spearman rank-order correlation is equal to the Pearson correlation between the rank values of the two variables, thereby also ranging between -1 and 1. Detecting the power of the Spearman rank correlation test is an important topic in the analysis of hydrological time series data.

The test uses Fisher's asymptotic method to estimate the power for the one-sample Spearman rank-order correlation.

Partial

Partial correlation can be explained as the association between two random variables after eliminating the effect of another or several other variables. It is a useful measurement in the presence of confounding. Similar to the Pearson correlation coefficient, partial correlation coefficient is also a dimensionless quantity ranging between -1 and 1.

The test uses Fisher's asymptotic method to estimate the power for the one-sample Pearson correlation.

Regression

Univariate Linear

Univariate linear regression is a basic and standard statistical approach in which researchers use the values of several variables to explain or predict values of a scale outcome.

The test invokes power analysis for the type III *F*-test in univariate linear regression.

Command enhancements

CORRELATIONS command

Added support for the **`FULL`** and **`LOWER`** keywords in the **`PRINT`** subcommand. The keywords control the display of the correlation matrix table's lower triangle or the full correlation matrix table. The keywords were introduced to allow table output to adhere to APA style guidelines.

MATRIX-END MATRIX command

- The **`NCDF.BETA`** cumulative distribution function is now supported.
- Probability density functions are now supported (they were previously only supported by the **`COMPUTE`** command).
- Tail probability functions are now supported (they were previously only supported by the **`COMPUTE`** command).
- Random variable functions are now supported (they were previously only supported by the **`COMPUTE`** command).

NONPAR CORR command

Added support for the **`FULL`** and **`LOWER`** keywords in the **`PRINT`** subcommand. The keywords control the display of the correlation matrix table's lower triangle or the full correlation matrix table. The keywords were introduced to allow table output to adhere to APA style guidelines.

NPTESTS command

CRITERIA subcommand

The **`SEED`** keyword is now supported. The keyword resets the random seed used for the Monte Carlo sampling.

ONESAMPLE subcommand

The **`KOLMOGOROV_SMIRNOV`** keyword now supports the following Lilliefors test for Monte Carlo sampling settings:

NSAMPLES keyword

Resets the number of replicates used by the Lilliefors test for Monte Carlo sampling.

MC_CILEVEL keyword

Resets the confidence interval level that is estimated by the Kolmogorov-Smirnov test.

SIMULATION keyword

Controls whether the Monte Carlo simulation will be used to conduct the Lilliefors test for Normal distribution when the parameters are not specified.

POISSON keyword

The **`SAMPLE`** setting has been removed from the **`POISSON`** keyword.

NPAR TESTS command

KS_SIM subcommand

The **`KS_SIM`** subcommand is now supported. **`KS_SIM`** (KOLMOGOROV-SMIRNOV simulation) controls the parameters for the Monte Carlo simulation for Normal, Uniform, and Exponential distributions. The new subcommand supports the following Lilliefors test for Monte Carlo sampling keywords:

CIN keyword

Resets the estimated confidence interval level used by the Kolmogorov-Smirnov test (using the Monte Carlo simulations).

SAMPLES keyword

Resets the number of replicates used by the Lilliefors test for Monte Carlo sampling.

NONORMAL keyword

When specified, results will not include the Monte Carlo sampling for Normal distribution.

K-S subcommand

`POISSON=varlist` is no longer supported.

ONEWAY command

The **`CRITERIA`** and **`ES`** subcommands are now supported by the **`ONEWAY`** command:

CRITERIA subcommand

The optional subcommand controls the significance level to estimate the confidence intervals.

ES subcommand

The optional subcommand controls the effect size estimation by providing keywords for controlling the effect size calculation for the overall test, and controlling the calculation of the contrast test effect size.

OUTPUT MODIFY

- Added support for the **`PIVOT`** keyword in the **`TABLES`** subcommand. The keyword pivots the specified row dimension to the specified column dimension. Any existing column dimensions are incremented outwards. The keyword was introduced to allow table output to adhere to APA style guidelines.
- Added support for the **`HIDE`** and **`UNGROUP`** keywords in the **`TABLECELLS`** subcommand. **`HIDE`** suppresses the selected row or column; **`UNGROUP`** deletes the selected row or column group header. The keywords were introduced to allow table output to adhere to APA style guidelines.
- Added support for the **`PARENT`** and **`CHILD`** options for the **`SELECTCONDITION`** keyword in the **`TABLECELLS`** subcommand. Both options specify primary and secondary string conditions to apply changes within the area of the table specified by the SELECT keyword.

POWER ONEWAY ANOVA command

The new command estimates the power for two types of hypothesis to compare the multiple group means, the overall test, and the test with specified contrasts. The over test focuses on the null hypothesis that all group means are equal. The test with specified contrasts breaks down the overall ANOVA hypotheses into smaller but more describable and useful pieces of the means.

POWER MEANS INDEPENDENT command

The new command invokes power analysis for the independent sample *t*-test to draw statistical inference about the difference of the two means.

POWER MEANS ONESAMPLE command

The new command invokes power analysis for the one sample *t*-test to draw statistical inference about the mean parameter.

POWER MEANS RELATED command

The new command invokes power analysis for the related sample *t*-test to draw statistical inference about the difference of the two means.

POWER PARTIALCORR command

The new command invokes the power analysis for the one-sample partial correlation test. Partial correlation can be explained as the association between two random variables after eliminating the effect of another or several other variables. It is a useful measurement in the presence of confounding.

POWER PEARSON ONESAMPLE command

The new command invokes the power analysis for the one-sample Pearson correlation test. Pearson product-moment correlation coefficient measures the strength of linear association between two scale random variables, which are assumed to follow a bivariate normal distribution.

POWER PROPORTIONS INDEPENDENT command

The new command invokes the power analysis for the independent-sample binomial test to compare two independent proportion parameters.

POWER PROPORTIONS ONESAMPLE command

The new command invokes power analysis for the one-sample binomial test to make statistical inference about the proportion parameter by comparing it with a hypothesized value.

POWER PROPORTIONS RELATED command

The new command invokes power analysis for the related-sample binomial test (or McNemar's test) to compare two proportion parameters based on the matched pair subjects sampled from two related binomial populations.

POWER SPEARMAN ONESAMPLE command

The new command invokes the power analysis for the one-sample Spearman rank-order correlation test. Spearman rank-order correlation coefficient is a rank-based nonparametric statistic to measure the monotonic relationship between two variables that are usually censored and not normally distributed.

POWER UNIVARIATE LINEAR command

The new command invokes power analysis for the type III *F*-test in univariate linear regression. Univariate linear regression is a basic and standard statistical approach in which researchers use the values of several variables to explain or predict values of a scale outcome.

QUANTILE REGRESSION command

CRITERIA subcommand

The **`QUANTILE`** keyword now provides support for a grid of quantiles (connected by the keywords TO and BY). The quantile grid can be mixed with other quantiles, and can be placed anywhere.

T-TEST command

The **ES** subcommand is now supported:

ES subcommand

The optional subcommand controls the effect size estimation by providing keywords for controlling the printing of the effect size calculation for the overall test, and controlling how the standardizer is computed in estimating the Cohen's *d* and Hedges' correction for each variable pair (only for Paired-Samples T Test).

WEIGHTED KAPPA command

Cohen’s kappa statistic is broadly used in cross-classification as a measure of agreement between two observed raters. It is an appropriate index of agreement when ratings are nominal scales with no order structure. The new `WEIGHTED KAPPA` command is an important generalization of the kappa statistic that measures the agreement of two ordinal subjects with identical categories.

Charting enhancements

Chart Builder has been updated to include the following features/enhancements.

Chart templates

- The **Edit** > **Options** > **Chart** dialog now includes a Samples Settings section that provides preview settings for any selected chart template. The dialog dynamically updates the preview chart images based on the specified settings.

Default chart colors
The default chart colors have been changed to a purple theme.

Chart Builder > Chart Appearance tab
The tab now allows you to directly select different chart template files.

Chart Editor
You can now increase/decrease font sizes directly in the editor.

High resolution DPI export
When the **Graphics only** option is selected as the document type in the Objects to Export dialog, the default file type is now set to **Production Ready Postscript (*.eps)**, which is a high resolution image format.

Legends and titles
You can now move chart images and titles directly in the output.

Search enhancements
The Search feature has been updated to provide results that include:

- Menu dialogs
- Help topics
- Case studies
- Syntax reference

Clicking a search result will take you directly to the relevant procedure dialog, help topic, case study, or syntax reference topic.

Chapter 2. Analyze procedures

Bivariate Correlations

The Bivariate Correlations procedure computes Pearson's correlation coefficient, Spearman's rho, and Kendall's tau-*b* with their significance levels. Correlations measure how variables or rank orders are related. Before calculating a correlation coefficient, screen your data for outliers (which can cause misleading results) and evidence of a linear relationship. Pearson's correlation coefficient is a measure of linear association. Two variables can be perfectly related, but if the relationship is not linear, Pearson's correlation coefficient is not an appropriate statistic for measuring their association.

Example

Is the number of games won by a basketball team correlated with the average number of points scored per game? A scatterplot indicates that there is a linear relationship. Analyzing data from the 1994–1995 NBA season yields that Pearson's correlation coefficient (0.581) is significant at the 0.01 level. You might suspect that the more games won per season, the fewer points the opponents scored. These variables are negatively correlated (–0.401), and the correlation is significant at the 0.05 level.

Statistics

For each variable: number of cases with nonmissing values, mean, and standard deviation. For each pair of variables: Pearson's correlation coefficient, Spearman's rho, Kendall's tau-*b*, cross-product of deviations, and covariance.

Data considerations

Data

Use symmetric quantitative variables for Pearson's correlation coefficient and quantitative variables or variables with ordered categories for Spearman's rho and Kendall's tau-*b*.

Assumptions

Pearson's correlation coefficient assumes that each pair of variables is bivariate normal.

Obtaining Bivariate Correlations

From the menus choose:

Analyze > **Correlate** > **Bivariate...**

1. Select two or more numeric variables.

The following options are also available:

Correlation Coefficients

For quantitative, normally distributed variables, choose the **Pearson** correlation coefficient. If your data are not normally distributed or have ordered categories, choose **Kendall's tau-b** or **Spearman**, which measure the association between rank orders. Correlation coefficients range in value from –1 (a perfect negative relationship) and +1 (a perfect positive relationship). A value of 0 indicates no linear relationship. When interpreting your results, be careful not to draw any cause-and-effect conclusions due to a significant correlation.

Test of Significance

You can select two-tailed or one-tailed probabilities. If the direction of association is known in advance, select **One-tailed**. Otherwise, select **Two-tailed**.

Flag significant correlations

Correlation coefficients significant at the 0.05 level are identified with a single asterisk, and those significant at the 0.01 level are identified with two asterisks.

Show only the lower triangle

When selected, only the correlation matrix table's lower triangle is presented in the output. When not selected, the full correlation matrix table is presented in the output. The setting allows table output to adhere to APA style guidelines.

Bivariate Correlations Options

Statistics. For Pearson correlations, you can choose one or both of the following:

- **Means and standard deviations.** Displayed for each variable. The number of cases with nonmissing values is also shown. Missing values are handled on a variable-by-variable basis regardless of your missing values setting.
- **Cross-product deviations and covariances.** Displayed for each pair of variables. The cross-product of deviations is equal to the sum of the products of mean-corrected variables. This is the numerator of the Pearson correlation coefficient. The covariance is an unstandardized measure of the relationship between two variables, equal to the cross-product deviation divided by N–1.

Missing Values. You can choose one of the following:

- **Exclude cases pairwise.** Cases with missing values for one or both of a pair of variables for a correlation coefficient are excluded from the analysis. Since each coefficient is based on all cases that have valid codes on that particular pair of variables, the maximum information available is used in every calculation. This can result in a set of coefficients based on a varying number of cases.
- **Exclude cases listwise.** Cases with missing values for any variable are excluded from all correlations.

CORRELATIONS and NONPAR CORR Command Additional Features

The command syntax language also allows you to:

- Write a correlation matrix for Pearson correlations that can be used in place of raw data to obtain other analyses such as factor analysis (with the `MATRIX` subcommand).
- Obtain correlations of each variable on a list with each variable on a second list (using the keyword `WITH` on the `VARIABLES` subcommand).

See the *Command Syntax Reference* for complete syntax information.

Crosstabs

The Crosstabs procedure forms two-way and multiway tables and provides a variety of tests and measures of association for two-way tables. The structure of the table and whether categories are ordered determine what test or measure to use.

Crosstabs' statistics and measures of association are computed for two-way tables only. If you specify a row, a column, and a layer factor (control variable), the Crosstabs procedure forms one panel of associated statistics and measures for each value of the layer factor (or a combination of values for two or more control variables). For example, if *gender* is a layer factor for a table of *married* (yes, no) against *life* (is life exciting, routine, or dull), the results for a two-way table for the females are computed separately from those for the males and printed as panels following one another.

Example. Are customers from small companies more likely to be profitable in sales of services (for example, training and consulting) than those from larger companies? From a crosstabulation, you might learn that the majority of small companies (fewer than 500 employees) yield high service profits, while the majority of large companies (more than 2,500 employees) yield low service profits.

Statistics and measures of association. Pearson chi-square, likelihood-ratio chi-square, linear-by-linear association test, Fisher's exact test, Yates' corrected chi-square, Pearson's *r*, Spearman's rho, contingency coefficient, phi, Cramér's *V*, symmetric and asymmetric lambdas, Goodman and Kruskal's tau, uncertainty coefficient, gamma, Somers' *d*, Kendall's tau-*b*, Kendall's tau-*c*, eta coefficient, Cohen's kappa, relative risk estimate, odds ratio, McNemar test, Cochran's and Mantel-Haenszel statistics, and column proportions statistics.

Crosstabs Data Considerations

Data. To define the categories of each table variable, use values of a numeric or string (eight or fewer bytes) variable. For example, for *gender*, you could code the data as 1 and 2 or as *male* and *female*.

Assumptions. Some statistics and measures assume ordered categories (ordinal data) or quantitative values (interval or ratio data), as discussed in the section on statistics. Others are valid when the table variables have unordered categories (nominal data). For the chi-square-based statistics (phi, Cramér's *V*, and contingency coefficient), the data should be a random sample from a multinomial distribution.

Note: Ordinal variables can be either numeric codes that represent categories (for example, 1 = *low*, 2 = *medium*, 3 = *high*) or string values. However, the alphabetic order of string values is assumed to reflect the true order of the categories. For example, for a string variable with the values of *low*, *medium*, *high*, the order of the categories is interpreted as *high*, *low*, *medium*--which is not the correct order. In general, it is more reliable to use numeric codes to represent ordinal data.

To Obtain Crosstabulations

1. From the menus choose:

   Analyze > **Descriptive Statistics** > **Crosstabs...**
2. Select one or more row variables and one or more column variables.

Optionally, you can:

- Select one or more control variables.
- Click **Statistics** for tests and measures of association for two-way tables or subtables.
- Click **Cells** for observed and expected values, percentages, and residuals.
- Click **Format** for controlling the order of categories.

Crosstabs layers

If you select one or more layer variables, a separate crosstabulation is produced for each category of each layer variable (control variable). For example, if you have one row variable, one column variable, and one layer variable with two categories, you get a two-way table for each category of the layer variable. To make another layer of control variables, click **Next**. Subtables are produced for each combination of categories for each first-layer variable, each second-layer variable, and so on. If statistics and measures of association are requested, they apply to two-way subtables only.

Crosstabs clustered bar charts

Display clustered bar charts. A clustered bar chart helps summarize your data for groups of cases. There is one cluster of bars for each value of the variable you specified under Rows. The variable that defines the bars within each cluster is the variable you specified under Columns. There is one set of differently colored or patterned bars for each value of this variable. If you specify more than one variable under Columns or Rows, a clustered bar chart is produced for each combination of two variables.

Crosstabs displaying layer variables in table layers

Display layer variables in table layers. You can choose to display the layer variables (control variables) as table layers in the crosstabulation table. This allows you to create views that show the overall statistics for row and column variables as well as permitting drill down on categories of layer variables.

An example that uses the data file *demo.sav* (available in the Samples directory of the installation directory) is shown below and was obtained as follows:

1. Select *Income category in thousands (inccat)* as the row variable, *Owns PDA (ownpda)* as the column variable and *Level of Education (ed)* as the layer variable.
2. Select **Display layer variables in table layers**.
3. Select **Column** in the Cell Display subdialog.

4. Run the Crosstabs procedure, double-click the crosstabulation table and select **College degree** from the Level of education drop down list.

The selected view of the crosstabulation table shows the statistics for respondents who have a college degree.

Crosstabs statistics

Chi-square. For tables with two rows and two columns, select **Chi-square** to calculate the Pearson chi-square, the likelihood-ratio chi-square, Fisher's exact test, and Yates' corrected chi-square (continuity correction). For 2 × 2 tables, Fisher's exact test is computed when a table that does not result from missing rows or columns in a larger table has a cell with an expected frequency of less than 5. Yates' corrected chi-square is computed for all other 2 × 2 tables. For tables with any number of rows and columns, select **Chi-square** to calculate the Pearson chi-square and the likelihood-ratio chi-square. When both table variables are quantitative, **Chi-square** yields the linear-by-linear association test.

Correlations. For tables in which both rows and columns contain ordered values, **Correlations** yields Spearman's correlation coefficient, rho (numeric data only). Spearman's rho is a measure of association between rank orders. When both table variables (factors) are quantitative, **Correlations** yields the Pearson correlation coefficient, *r*, a measure of linear association between the variables.

Nominal. For nominal data (no intrinsic order, such as Catholic, Protestant, and Jewish), you can select **Contingency coefficient**, **Phi** (coefficient) **and Cramér's V**, **Lambda** (symmetric and asymmetric lambdas and Goodman and Kruskal's tau), and **Uncertainty coefficient**.

- *Contingency coefficient*. A measure of association based on chi-square. The value ranges between 0 and 1, with 0 indicating no association between the row and column variables and values close to 1 indicating a high degree of association between the variables. The maximum value possible depends on the number of rows and columns in a table.
- *Phi and Cramer's V*. Phi is a chi-square-based measure of association that involves dividing the chi-square statistic by the sample size and taking the square root of the result. Cramer's V is a measure of association based on chi-square.
- *Lambda*. A measure of association that reflects the proportional reduction in error when values of the independent variable are used to predict values of the dependent variable. A value of 1 means that the independent variable perfectly predicts the dependent variable. A value of 0 means that the independent variable is no help in predicting the dependent variable.
- *Uncertainty coefficient*. A measure of association that indicates the proportional reduction in error when values of one variable are used to predict values of the other variable. For example, a value of 0.83 indicates that knowledge of one variable reduces error in predicting values of the other variable by 83%. The program calculates both symmetric and asymmetric versions of the uncertainty coefficient.

Ordinal. For tables in which both rows and columns contain ordered values, select **Gamma** (zero-order for 2-way tables and conditional for 3-way to 10-way tables), **Kendall's tau-b**, and **Kendall's tau-c**. For predicting column categories from row categories, select **Somers' d**.

- *Gamma*. A symmetric measure of association between two ordinal variables that ranges between -1 and 1. Values close to an absolute value of 1 indicate a strong relationship between the two variables. Values close to 0 indicate little or no relationship. For 2-way tables, zero-order gammas are displayed. For 3-way to n-way tables, conditional gammas are displayed.
- *Somers' d*. A measure of association between two ordinal variables that ranges from -1 to 1. Values close to an absolute value of 1 indicate a strong relationship between the two variables, and values close to 0 indicate little or no relationship between the variables. Somers' d is an asymmetric extension of gamma that differs only in the inclusion of the number of pairs not tied on the independent variable. A symmetric version of this statistic is also calculated.
- *Kendall's tau-b*. A nonparametric measure of correlation for ordinal or ranked variables that take ties into account. The sign of the coefficient indicates the direction of the relationship, and its absolute value indicates the strength, with larger absolute values indicating stronger relationships. Possible values range from -1 to 1, but a value of -1 or +1 can be obtained only from square tables.

- *Kendall's tau-c*. A nonparametric measure of association for ordinal variables that ignores ties. The sign of the coefficient indicates the direction of the relationship, and its absolute value indicates the strength, with larger absolute values indicating stronger relationships. Possible values range from -1 to 1, but a value of -1 or +1 can be obtained only from square tables.

Nominal by Interval. When one variable is categorical and the other is quantitative, select **Eta**. The categorical variable must be coded numerically.

- *Eta*. A measure of association that ranges from 0 to 1, with 0 indicating no association between the row and column variables and values close to 1 indicating a high degree of association. Eta is appropriate for a dependent variable measured on an interval scale (for example, income) and an independent variable with a limited number of categories (for example, gender). Two eta values are computed: one treats the row variable as the interval variable, and the other treats the column variable as the interval variable.

Kappa. Cohen's kappa measures the agreement between the evaluations of two raters when both are rating the same object. A value of 1 indicates perfect agreement. A value of 0 indicates that agreement is no better than chance. Kappa is based on a square table in which row and column values represent the same scale. Any cell that has observed values for one variable but not the other is assigned a count of 0. Kappa is not computed if the data storage type (string or numeric) is not the same for the two variables. For string variable, both variables must have the same defined length.

Risk. For 2 x 2 tables, a measure of the strength of the association between the presence of a factor and the occurrence of an event. If the confidence interval for the statistic includes a value of 1, you cannot assume that the factor is associated with the event. The odds ratio can be used as an estimate or relative risk when the occurrence of the factor is rare.

McNemar. A nonparametric test for two related dichotomous variables. Tests for changes in responses using the chi-square distribution. Useful for detecting changes in responses due to experimental intervention in "before-and-after" designs. For larger square tables, the McNemar-Bowker test of symmetry is reported.

Cochran's and Mantel-Haenszel statistics. Cochran's and Mantel-Haenszel statistics can be used to test for independence between a dichotomous factor variable and a dichotomous response variable, conditional upon covariate patterns defined by one or more layer (control) variables. Note that while other statistics are computed layer by layer, the Cochran's and Mantel-Haenszel statistics are computed once for all layers.

Crosstabs cell display

To help you uncover patterns in the data that contribute to a significant chi-square test, the Crosstabs procedure displays expected frequencies and three types of residuals (deviates) that measure the difference between observed and expected frequencies. Each cell of the table can contain any combination of counts, percentages, and residuals selected.

Counts. The number of cases actually observed and the number of cases expected if the row and column variables are independent of each other. You can choose to hide counts that are less than a specified integer. Hidden values will be displayed as **<N**, where **N** is the specified integer. The specified integer must be greater than or equal to 2, although the value 0 is permitted and specifies that no counts are hidden.

Compare column proportions. This option computes pairwise comparisons of column proportions and indicates which pairs of columns (for a given row) are significantly different. Significant differences are indicated in the crosstabulation table with APA-style formatting using subscript letters and are calculated at the 0.05 significance level. *Note*: If this option is specified without selecting observed counts or column percentages, then observed counts are included in the crosstabulation table, with the APA-style subscript letters indicating the results of the column proportions tests.

- **Adjust p-values (Bonferroni method).** Pairwise comparisons of column proportions make use of the Bonferroni correction, which adjusts the observed significance level for the fact that multiple comparisons are made.

Percentages. The percentages can add up across the rows or down the columns. The percentages of the total number of cases represented in the table (one layer) are also available.

Note: If **Hide small counts** is selected in the Counts group, then percentages associated with hidden counts are also hidden.

Residuals. Raw unstandardized residuals give the difference between the observed and expected values. Standardized and adjusted standardized residuals are also available.

- *Unstandardized*. The difference between an observed value and the expected value. The expected value is the number of cases you would expect in the cell if there were no relationship between the two variables. A positive residual indicates that there are more cases in the cell than there would be if the row and column variables were independent.
- *Standardized*. The residual divided by an estimate of its standard deviation. Standardized residuals, which are also known as Pearson residuals, have a mean of 0 and a standard deviation of 1.
- *Adjusted standardized*. The residual for a cell (observed minus expected value) divided by an estimate of its standard error. The resulting standardized residual is expressed in standard deviation units above or below the mean.

Create APA style table. Creates output tables that adhere to APA style guidelines.

Note: The **Observed**, **Expected**, **Row**, **Column**, and **Total** options are not available when **Create APA style table** is selected.

Noninteger Weights. Cell counts are normally integer values, since they represent the number of cases in each cell. But if the data file is currently weighted by a weight variable with fractional values (for example, 1.25), cell counts can also be fractional values. You can truncate or round either before or after calculating the cell counts or use fractional cell counts for both table display and statistical calculations.

- *Round cell counts*. Case weights are used as is but the accumulated weights in the cells are rounded before computing any statistics.
- *Truncate cell counts*. Case weights are used as is but the accumulated weights in the cells are truncated before computing any statistics.
- *Round case weights*. Case weights are rounded before use.
- *Truncate case weights*. Case weights are truncated before use.
- *No adjustments*. Case weights are used as is and fractional cell counts are used. However, when Exact Statistics (available only with Sampling and Testing) are requested, the accumulated weights in the cells are either truncated or rounded before computing the Exact test statistics.

Crosstabs table format

You can arrange rows in ascending or descending order of the values of the row variable.

Frequencies

The Frequencies procedure provides statistics and graphical displays that are useful for describing many types of variables. The Frequencies procedure is a good place to start looking at your data.

For a frequency report and bar chart, you can arrange the distinct values in ascending or descending order, or you can order the categories by their frequencies. The frequencies report can be suppressed when a variable has many distinct values. You can label charts with frequencies (the default) or percentages.

Example. What is the distribution of a company's customers by industry type? From the output, you might learn that 37.5% of your customers are in government agencies, 24.9% are in corporations, 28.1% are in academic institutions, and 9.4% are in the healthcare industry. For continuous, quantitative data, such as sales revenue, you might learn that the average product sale is $3,576, with a standard deviation of $1,078.

Statistics and plots. Frequency counts, percentages, cumulative percentages, mean, median, mode, sum, standard deviation, variance, range, minimum and maximum values, standard error of the mean, skewness and kurtosis (both with standard errors), quartiles, user-specified percentiles, bar charts, pie charts, and histograms.

Frequencies Data Considerations

Data. Use numeric codes or strings to code categorical variables (nominal or ordinal level measurements).

Assumptions. The tabulations and percentages provide a useful description for data from any distribution, especially for variables with ordered or unordered categories. Most of the optional summary statistics, such as the mean and standard deviation, are based on normal theory and are appropriate for quantitative variables with symmetric distributions. Robust statistics, such as the median, quartiles, and percentiles, are appropriate for quantitative variables that may or may not meet the assumption of normality.

To Obtain Frequency Tables

1. From the menus choose:

   Analyze > **Descriptive Statistics** > **Frequencies...**
2. Select one or more categorical or quantitative variables.

Optionally, you can:

- Select **Create APA style tables** to create output tables that adhere to APA style guidelines.
- Click **Statistics** for descriptive statistics for quantitative variables.
- Click **Charts** for bar charts, pie charts, and histograms.
- Click **Format** for the order in which results are displayed.

Frequencies Statistics

Percentile Values. Values of a quantitative variable that divide the ordered data into groups so that a certain percentage is above and another percentage is below. Quartiles (the 25th, 50th, and 75th percentiles) divide the observations into four groups of equal size. If you want an equal number of groups other than four, select **Cut points for n equal groups.** You can also specify individual percentiles (for example, the 95th percentile, the value below which 95% of the observations fall).

Central Tendency. Statistics that describe the location of the distribution include the mean, median, mode, and sum of all the values.

- *Mean*. A measure of central tendency. The arithmetic average, the sum divided by the number of cases.
- *Median*. The value above and below which half of the cases fall, the 50th percentile. If there is an even number of cases, the median is the average of the two middle cases when they are sorted in ascending or descending order. The median is a measure of central tendency not sensitive to outlying values (unlike the mean, which can be affected by a few extremely high or low values).
- *Mode*. The most frequently occurring value. If several values share the greatest frequency of occurrence, each of them is a mode. The Frequencies procedure reports only the smallest of such multiple modes.
- *Sum*. The sum or total of the values, across all cases with nonmissing values.

Dispersion. Statistics that measure the amount of variation or spread in the data include the standard deviation, variance, range, minimum, maximum, and standard error of the mean.

- *Std. deviation*. A measure of dispersion around the mean. In a normal distribution, 68% of cases fall within one standard deviation of the mean and 95% of cases fall within two standard deviations. For example, if the mean age is 45, with a standard deviation of 10, 95% of the cases would be between 25 and 65 in a normal distribution.
- *Variance*. A measure of dispersion around the mean, equal to the sum of squared deviations from the mean divided by one less than the number of cases. The variance is measured in units that are the square of those of the variable itself.

- *Range*. The difference between the largest and smallest values of a numeric variable, the maximum minus the minimum.
- *Minimum*. The smallest value of a numeric variable.
- *Maximum*. The largest value of a numeric variable.
- *S. E. mean*. A measure of how much the value of the mean may vary from sample to sample taken from the same distribution. It can be used to roughly compare the observed mean to a hypothesized value (that is, you can conclude the two values are different if the ratio of the difference to the standard error is less than -2 or greater than +2).

Distribution. Skewness and kurtosis are statistics that describe the shape and symmetry of the distribution. These statistics are displayed with their standard errors.

- *Skewness*. A measure of the asymmetry of a distribution. The normal distribution is symmetric and has a skewness value of 0. A distribution with a significant positive skewness has a long right tail. A distribution with a significant negative skewness has a long left tail. As a guideline, a skewness value more than twice its standard error is taken to indicate a departure from symmetry.
- *Kurtosis*. A measure of the extent to which there are outliers. For a normal distribution, the value of the kurtosis statistic is zero. Positive kurtosis indicates that the data exhibit more extreme outliers than a normal distribution. Negative kurtosis indicates that the data exhibit less extreme outliers than a normal distribution.

Values are group midpoints. If the values in your data are midpoints of groups (for example, ages of all people in their thirties are coded as 35), select this option to estimate the median and percentiles for the original, ungrouped data.

Frequencies Charts

Chart Type. A pie chart displays the contribution of parts to a whole. Each slice of a pie chart corresponds to a group that is defined by a single grouping variable. A bar chart displays the count for each distinct value or category as a separate bar, allowing you to compare categories visually. A histogram also has bars, but they are plotted along an equal interval scale. The height of each bar is the count of values of a quantitative variable falling within the interval. A histogram shows the shape, center, and spread of the distribution. A normal curve superimposed on a histogram helps you judge whether the data are normally distributed.

Chart Values. For bar charts, the scale axis can be labeled by frequency counts or percentages.

Frequencies Format

Order by. The frequency table can be arranged according to the actual values in the data or according to the count (frequency of occurrence) of those values, and the table can be arranged in either ascending or descending order. However, if you request a histogram or percentiles, Frequencies assumes that the variable is quantitative and displays its values in ascending order.

Multiple Variables. If you produce statistics tables for multiple variables, you can either display all variables in a single table (**Compare variables**) or display a separate statistics table for each variable (**Organize output by variables)**.

Suppress tables with many categories. This option prevents the display of tables with more than the specified number of values.

Power Analysis

Power analysis plays a pivotal role in a study plan, design, and conduction. The calculation of power is usually before any sample data have been collected, except possibly from a small pilot study. The precise estimation of the power may tell investigators how likely it is that a statistically significant difference will be detected based on a finite sample size under a true alternative hypothesis. If the power is too low, there is little chance of detecting a significant difference, and non-significant results are likely even if real differences truly exist.

IBM SPSS Statistics provides the following Power Analysis procedures:

One Sample T-Test
In one-sample analysis, the observed data are collected as a single random sample. It is assumed that the sample data independently and identically follow a normal distribution with a fixed mean and variance, and draws statistical inference about the mean parameter.

Paired Sample T-Test
In paired-sample analysis, the observed data contain two paired and correlated samples, and each case has two measurements. It is assumed that the data in each sample independently and identically follow a normal distribution with a fixed mean and variance, and draws statistical inference about the difference of the two means.

Independent Sample T-Test
In independent-sample analysis, the observed data contain two independent samples. It is assumed that the data in each sample independently and identically follow a normal distribution with a fixed mean and variance, and draws statistical inference about the difference of the two means.

One-way ANOVA
Analysis of variance (ANOVA) is a statistical method of estimating the means of several populations which are often assumed to be normally distributed. The One-way ANOVA, a common type of ANOVA, is an extension of the two-sample *t*-test.

Example. The power of a hypothesis test to detect a correct alternative hypothesis is the probability that the test will reject the test hypothesis. Since the probability of a type II error is the probability of accepting the null hypothesis when the alternative hypothesis is true, the power can be expressed as (1-probability of a type II error), which is the probability of rejecting the null hypothesis when the alternative hypothesis is true.

Statistics and plots. One-sided test, two-sided test, significance level, Type I error rate, test assumptions, population standard deviation, population mean under testing, hypothesized value, two-dimensional power by sample size, two-dimensional power by effect size, three-dimensional power by sample size, three-dimensional power by effect size, rotation degrees, group pairs, Pearson product-moment correlation coefficient, mean difference, plot range of the effect size, pooled population standard deviation, contrasts and pairwise differences, contrast coefficients, contrast test, BONFERRONI, SIDAK, LSD, power by total sample size, two-dimensional power by pooled standard deviation, three-dimensional power by total sample, three-dimensional power by total sample size, pooled standard deviation, Student's t-distribution, non-central t-distribution,

Power Analysis data considerations

Data
In one-sample analysis, the observed data are collected as a single random sample.

In paired-sample analysis, the observed data contain two paired and correlated samples, and each case has two measurements.

In independent-sample analysis, the observed data contain two independent samples.

Analysis of variance (ANOVA) is a statistical method of estimating the means of several populations which are often assumed to be normally distributed.

Assumptions
In one-sample analysis, it is assumed that the sample data independently and identically follow a normal distribution with a fixed mean and variance, and draws statistical inference about the mean parameter.

In paired-sample analysis, it is assumed that the data in each sample independently and identically follow a normal distribution with a fixed mean and variance, and draws statistical inference about the difference of the two means.

In independent-sample analysis, it is assumed that the data in each sample independently and identically follow a normal distribution with a fixed mean and variance, and draws statistical inference about the difference of the two means.

In one-way ANOVA, the statistical method of estimating the means of several populations are often assumed to be normally distributed.

Obtaining a Power Analysis

1. From the menus choose:

   Analyze > **Power Analysis** > **Compare Means** > **One-Sample T-Test**, or **Paired-Sample T-Test**, or **Independent-Sample T-Test**, or **One-way ANOVA**
2. Define the required test assumptions.
3. Click **OK**.

Means

The following statistics features are included in IBM SPSS Statistics Base Edition.

Power Analysis of One-Sample T Test

This feature requires IBM SPSS Statistics Base Edition.

Power analysis plays a pivotal role in a study plan, design, and conduction. The calculation of power is usually before any sample data have been collected, except possibly from a small pilot study. The precise estimation of the power may tell investigators how likely it is that a statistically significant difference will be detected based on a finite sample size under a true alternative hypothesis. If the power is too low, there is little chance of detecting a significant difference, and non-significant results are likely even if real differences truly exist.

In one-sample analysis, the observed data are collected as a single random sample. It is assumed that the sample data independently and identically follow a normal distribution with a fixed mean and variance, and draws statistical inference about the mean parameter.

1. From the menus choose:

   Analyze > **Power Analysis** > **Means** > **One-Sample T Test**
2. Select a test assumption setting (**Estimate sample size** or **Estimate power**).
3. When selecting **Estimate power**, enter an appropriate **Sample size** for power estimation value. The value must be an integer greater than 1. When selecting **Estimate sample size**, enter an appropriate **Power** for sample size estimation value. The value must be a single value between 0 and 1.
4. Enter a value that specifies the population mean under testing in the **Population mean** field. The value must be a single numeric.
5. Optionally, enter a value that specifies the null hypothesis value to be tested in the **Null value** field. The value must be a single numeric.
6. Enter a **Population standard deviation** value. The value must be a single numeric greater than 0.
7. Select whether the test is one or two-sided.

   Nondirectional or two-sided analysis
   When selected, a two-sided test is used. This is the default setting.

   Directional or one-sided analysis
   When selected, power is computed for a one-sided test.
8. Optionally, specify the significance level of the Type I error rate for the test in the **Significance level** field. The value must be a single double value between 0 and 1. The default value is 0.05.
9. You can optionally click **Plot** to specify "Power Analysis of One-Sample T Test: Plot" on page 17 settings (chart output, two-dimensional plot settings, three-dimensional plot settings, and tooltips).

   Note: Plot is available only when **Estimate power** is selected as the test assumption.

Power Analysis of One-Sample T Test: Plot

You can control the plots that are output to illustrate the two and three-dimensional power by sample/effect size charts. You can also control the display of tool tips and the vertical/horizontal rotation degrees for three-dimensional charts.

Two-Dimensional Plot

Power estimation versus sample size
When enabled, this optional setting provides options for controlling the two-dimensional power by sample size chart. The setting is disabled by default.

Range of sample size
When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range of the sample size is used.

Lower bound
Controls the lower bound for the two-dimensional power by sample size chart. The value must be greater than 1, and cannot be greater than the **Upper bound** value.

Upper bound
Controls the upper bound for the two-dimensional power by sample size chart. The value must be greater than the **Lower bound** value and cannot be greater than 5000.

Power estimation versus effect size
By default, this optional setting is disabled. When enabled, the chart displays in the output. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range of the effect size used.

Range of effect size
When selected, the lower and upper bound options are available.

Lower bound
Controls the lower bound for the two-dimensional power by effect size chart. The value must be greater than, or equal to, -5.0 and cannot be greater than the **Upper bound** value.

Upper bound
Controls the upper bound for the two-dimensional power by effect size chart. The value must be greater than the **Lower bound** value and cannot be greater than 5.0.

Three-Dimensional Plot

Power estimation versus
Provides options for controlling the three-dimensional power by sample size (x-axis) and effect size (y-axis) chart, the vertical/horizontal rotation settings, and the user specified plot range of sample/effect size. This setting is disabled by default.

Effect size on x-axis and sample size on y-axis
The optional setting controls the three-dimensional power by sample size (x-axis) and effect size (y-axis) chart. By default, the chart is suppressed. When specified, the chart displays.

Effect size on y-axis and sample size on x-axis
The optional setting controls the three-dimensional power by sample size (y-axis) and effect size (x-axis) chart. By default, the chart is suppressed. When specified, the chart displays.

Vertical rotation
The optional setting sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. You can use the mouse to rotate the chart vertically. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 10.

Horizontal rotation
The optional setting sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. You can use the mouse to rotate the chart horizontally. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 325.

Range of sample size

When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range of the sample size is used.

Lower bound

Controls the lower bound for the three-dimensional power by sample size chart. The value must be greater than 1, and cannot be greater than the **Upper bound** value.

Upper bound

Controls the upper bound for the three-dimensional power by sample size chart. The value must be greater than the **Lower bound** value and cannot be greater than 5000.

Range of effect size

When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range of the effect size is used.

Lower bound

Controls the lower bound for the three-dimensional power by effect size chart. The value must be greater than, or equal to, -5.0 and cannot be greater than the **Upper bound** value.

Upper bound

Controls the upper bound for the three-dimensional power by effect size chart. The value must be greater than the **Lower bound** value and cannot be greater than 5.0.

Power Analysis of Paired-Samples T Test

This feature requires IBM SPSS Statistics Base Edition.

Power analysis plays a pivotal role in a study plan, design, and conduction. The calculation of power is usually before any sample data have been collected, except possibly from a small pilot study. The precise estimation of the power may tell investigators how likely it is that a statistically significant difference will be detected based on a finite sample size under a true alternative hypothesis. If the power is too low, there is little chance of detecting a significant difference, and non-significant results are likely even if real differences truly exist.

In paired-samples analysis, the observed data contain two paired and correlated samples, and each case has two measurements. It is assumed that the data in each sample independently and identically follow a normal distribution with a fixed mean and variance, and draws statistical inference about the difference of the two means.

1. From the menus choose:

   Analyze > **Power Analysis** > **Means** > **Paired-Samples T Test**
2. Select a test assumption setting (**Estimate sample size** or **Estimate power**).
3. When selecting **Estimate power**, enter an appropriate **Sample size** for power estimation value. The value must be an integer greater than 1. When selecting **Estimate sample size**, enter an appropriate **Power** for sample size estimation value. The value must be a single value between 0 and 1.
4. When a single population mean is required, enter a **Population mean difference** value. When single value is specified, it denotes the population mean difference μ_d.

   Note: The value cannot be 0 when **Estimate sample size** is selected.
5. When multiple population means are required for the specified group pairs, enter values for **Population mean for group 1** and **Population mean for group 2**. When multiple values are specified, they denote the population mean difference μ_1 and μ_2.

   Note: The two values cannot be the same when **Estimate sample size** is selected.
6. When a single population mean is specified, enter the **Population standard deviation for mean difference** value. When a single value is specified, it denotes the population standard deviation of the group difference σ_d. The value must be a single numeric greater than 0.

7. When multiple population means are specified, enter the **Population standard deviation for group 1** and **Population standard deviation for group 2** values. When multiple values are specified, they denote the population standard deviation of the group difference σ_1 and σ_2. The values must be a single numerics greater than 0.
8. Optionally, enter a value that specifies the **Pearson product-moment correlation coefficient** ρ. The value must be a single numeric value between -1 and 1. The value cannot be 0.

   Note: When a single **Population standard deviation for mean difference** value is specified, this setting is ignored. Otherwise, the values for **Population standard deviation for group 1** and **Population standard deviation for group 2** are used to compute σ_d.
9. Select whether the test is one or two-sided.

   Nondirectional or two-sided analysis
   When selected, a two-sided test is used. This is the default setting.

   Directional or one-sided analysis
   When selected, power is computed for a one-sided test.
10. Optionally, specify the significance level of the Type I error rate for the test in the **Significance level** field. The value must be a single double value between 0 and 1. The default value is 0.05.
11. You can optionally click **Plot** to specify "Power Analysis of Paired-Samples T Test: Plot" on page 19 settings (chart output, two-dimensional plot settings, three-dimensional plot settings, and tooltips).

    Note: Plot is available only when **Estimate power** is selected as the test assumption.

Power Analysis of Paired-Samples T Test: Plot

You can control the plots that are output to illustrate the two and three-dimensional power by sample/effect size charts. You can also control the display of tool tips and the vertical/horizontal rotation degrees for three-dimensional charts.

Two-Dimensional Plot

Power estimation versus sample size
When enabled, this optional setting provides options for controlling the two-dimensional power by sample size chart. The setting is disabled by default.

Range of sample size
When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range of the sample size is used.

Lower bound
Controls the lower bound for the two-dimensional power by sample size chart. The value must be greater than 1, and cannot be greater than the **Upper bound** value.

Upper bound
Controls the upper bound for the two-dimensional power by sample size chart. The value must be greater than the **Lower bound** value and cannot be greater than 5000.

Power estimation versus effect size
By default, this optional setting is disabled. When enabled, the chart displays in the output. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range of the effect size used.

Range of effect size
When selected, the lower and upper bound options are available.

Lower bound
Controls the lower bound for the two-dimensional power by effect size chart. The value must be greater than, or equal to, -5.0 and cannot be greater than the **Upper bound** value.

Upper bound
Controls the upper bound for the two-dimensional power by effect size chart. The value must be greater than the **Lower bound** value and cannot be greater than 5.0.

Three-Dimensional Plot

Power estimation versus

Provides options for controlling the three-dimensional power by sample size (x-axis) and effect size (y-axis) chart, the vertical/horizontal rotation settings, and the user specified plot range of sample/effect size. This setting is disabled by default.

Effect size on x-axis and sample size on y-axis

The optional setting controls the three-dimensional power by sample size (x-axis) and effect size (y-axis) chart. By default, the chart is suppressed. When specified, the chart displays.

Effect size on y-axis and sample size on x-axis

The optional setting controls the three-dimensional power by sample size (y-axis) and effect size (x-axis) chart. By default, the chart is suppressed. When specified, the chart displays.

Vertical rotation

The optional setting sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. You can use the mouse to rotate the chart vertically. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 10.

Horizontal rotation

The optional setting sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. You can use the mouse to rotate the chart horizontally. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 325.

Range of sample size

When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range of the sample size is used.

Lower bound

Controls the lower bound for the three-dimensional power by sample size chart. The value must be greater than 1, and cannot be greater than the **Upper bound** value.

Upper bound

Controls the upper bound for the three-dimensional power by sample size chart. The value must be greater than the **Lower bound** value and cannot be greater than 5000.

Range of effect size

When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range of the effect size is used.

Lower bound

Controls the lower bound for the three-dimensional power by effect size chart. The value must be greater than, or equal to, -5.0 and cannot be greater than the **Upper bound** value.

Upper bound

Controls the upper bound for the three-dimensional power by effect size chart. The value must be greater than the **Lower bound** value and cannot be greater than 5.0.

Power Analysis of Independent-Samples T Test

This feature requires IBM SPSS Statistics Base Edition.

Power analysis plays a pivotal role in a study plan, design, and conduction. The calculation of power is usually before any sample data have been collected, except possibly from a small pilot study. The precise estimation of the power may tell investigators how likely it is that a statistically significant difference will be detected based on a finite sample size under a true alternative hypothesis. If the power is too low, there is little chance of detecting a significant difference, and non-significant results are likely even if real differences truly exist.

In independent-samples analysis, the observed data contain two independent samples. It is assumed that the data in each sample independently and identically follow a normal distribution with a fixed mean and variance, and draws statistical inference about the difference of the two means.

1. From the menus choose:

   Analyze > **Power Analysis** > **Means** > **Independent-Samples T Test**
2. Select a test assumption setting (**Estimate sample size** or **Estimate power**).
3. When **Estimate sample size** is selected, enter an appropriate **Power** for sample size estimation value (the value must be a single value between 0 and 1) and a **Group size ratio** value for specifying the ratio of the sample sizes (the value must be a single value between 0.01 and 100.
4. When **Estimate power** is selected, enter values to specify the sample size for the two groups for comparison **Sample size for group 1** and **Sample size for group 2**. The values must be an integers greater than 1.
5. When a single population mean is required, enter a **Population mean difference** value. When single value is specified, it denotes the population mean difference μ_d.

   Note: The value cannot be 0 when **Estimate sample size** is selected.
6. When multiple population means are required for the specified group pairs, enter values for **Population mean for group 1** and **Population mean for group 2**. When multiple values are specified, they denote the population mean difference μ_1 and μ_2.

   Note: The two values cannot be the same when **Estimate sample size** is selected.
7. Specify whether the population standard deviations are **Equal for two groups** or **Not equal for two groups**.

   - When the population standard deviations are equal for two groups, enter a value for **Pooled standard deviation** that denotes σ, and assumes that the two group variances are equal, or $\sigma_1 = \sigma_2 = \sigma$.
   - When the population standard deviations are not equal for two groups, enter values for **Standard deviation for group 1** and **Standard deviation for group 2** that denote σ_1 and σ_2.

     Note: When the values for **Standard deviation for group 1** and **Standard deviation for group 2** are identical, they are treated as a single value.
8. Select whether the test is one or two-sided.

   Nondirectional or two-sided analysis
   When selected, a two-sided test is used. This is the default setting.

   Directional or one-sided analysis
   When selected, power is computed for a one-sided test.
9. Optionally, specify the significance level of the Type I error rate for the test in the **Significance level** field. The value must be a single double value between 0 and 1. The default value is 0.05.
10. You can optionally click **Plot** to specify "Power Analysis of Independent-Samples T Test: Plot" on page 21 settings (chart output, two-dimensional plot settings, three-dimensional plot settings, and tooltips).

    Note: Plot is available only when **Estimate power** is selected as the test assumption.

Power Analysis of Independent-Samples T Test: Plot

You can control the plots that are output to illustrate the two and three-dimensional power by sample ratio, effect size, or mean difference charts. You can also control the display of tool tips and the vertical/ horizontal rotation degrees for three-dimensional charts.

Two-Dimensional Plot

Power estimation versus sample size ratio
When enabled, this optional setting provides options for controlling the two-dimensional power by sample size ratio chart. The setting is disabled by default.

Range of sample size ratio

When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range of the sample size is used.

Lower bound

Controls the lower bound for the two-dimensional power by sample size ratio chart. The value must be between 0.01 and 100 and cannot be greater than the **Upper bound** value.

Upper bound

Controls the upper bound for the two-dimensional power by sample size ratio chart. The value must be between 0.01 and 100 and must be greater than the **Lower bound** value.

Power estimation versus effect size (or mean difference)

By default, this optional setting is disabled. When enabled, the chart displays in the output. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range of the effect size (or mean difference) is used.

Range of effect size (or mean difference)

When selected, the lower and upper bound options are available.

Lower bound

Controls the lower bound for the two-dimensional power by effect size chart. The value must be greater than, or equal to, -5.0 and cannot be greater than the **Upper bound** value.

Upper bound

Controls the upper bound for the two-dimensional power by effect size chart. The value must be greater than the **Lower bound** value and cannot be greater than 5.0.

Three-Dimensional Plot

Power estimation versus

Provides options for controlling the three-dimensional power by sample size ratio (x-axis) and effect size (y-axis) chart, the vertical/horizontal rotation settings, and the user specified plot range of sample/effect size. This setting is disabled by default.

Effect size (or mean difference) on x-axis and sample size on y-axis

The optional setting controls the three-dimensional power by sample size ratio (x-axis) and effect size (y-axis) chart. By default, the chart is suppressed. When specified, the chart displays.

Effect size (or mean difference) on y-axis and sample size on x-axis

The optional setting controls the three-dimensional power by sample size (y-axis) and effect size (x-axis) chart. By default, the chart is suppressed. When specified, the chart displays.

Vertical rotation

The optional setting sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. You can use the mouse to rotate the chart vertically. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 10.

Horizontal rotation

The optional setting sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. You can use the mouse to rotate the chart horizontally. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 325.

Range of sample size ratio

When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range of the sample size is used.

Lower bound

Controls the lower bound for the three-dimensional power by sample size ratio chart. The value must be between 0.01 and 100 and cannot be greater than the **Upper bound** value.

Upper bound
Controls the upper bound for the three-dimensional power by sample size ratio chart. The value must be between 0.01 and 100 and must be greater than the **Lower bound** value.

Range of effect size (or mean difference)
When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range of the effect size is used.

Lower bound
Controls the lower bound for the three-dimensional power by effect size chart. The value must be greater than, or equal to, -5.0 and cannot be greater than the **Upper bound** value.

Upper bound
Controls the upper bound for the three-dimensional power by effect size chart. The value must be greater than the **Lower bound** value and cannot be greater than 5000.

Power Analysis of One-Way ANOVA

This feature requires IBM SPSS Statistics Base Edition.

Power analysis plays a pivotal role in a study plan, design, and conduction. The calculation of power is usually before any sample data have been collected, except possibly from a small pilot study. The precise estimation of the power may tell investigators how likely it is that a statistically significant difference will be detected based on a finite sample size under a true alternative hypothesis. If the power is too low, there is little chance of detecting a significant difference, and non-significant results are likely even if real differences truly exist.

Analysis of variance (ANOVA) is a statistical method of estimating the means of several populations which are often assumed to be normally distributed. The One-way ANOVA, a common type of ANOVA, is an extension of the two-sample *t*-test. The procedure provides approaches for estimating the power for two types of hypothesis to compare the multiple group means, the overall test, and the test with specified contrasts. The over test focuses on the null hypothesis that all group means are equal. The test with specified contrasts breaks down the overall ANOVA hypotheses into smaller but more describable and useful pieces of the means.

1. From the menus choose:

   Analyze > **Power Analysis** > **Means** > **One-way ANOVA**
2. Select a test assumption setting (**Estimate sample size** or **Estimate power**).
3. When **Estimate sample size** is selected, enter an appropriate **Power** for sample size estimation value (the value must be a single value between 0 and 1).
4. Enter a **Pooled population standard deviation** value. The value must be a single numeric greater than 0.
5. Specify the **Group sizes** and **Group means** values. At least two group size values must be specified (each value must be less than, or equal to, 2). At least two group mean values must also be specified (the number of specified values must equal the group size values).
6. Optionally, specify **Group weights** values. Group weights assign the group size weights when **Estimate power** is selected.

   Note: The **Group weights** settings are ignored when **Group sizes** values are specified.
7. Optionally, specify the significance level of the Type I error rate for the test in the **Significance level** field. The value must be a single double value between 0 and 1. The default value is 0.05.
8. You can optionally click **Contrast** to specify "Power Analysis of One-way ANOVA: Contrast" on page 24 settings (contrast test and pairwise differences), or click **Plot** to specify "Power Analysis of One-way ANOVA: Plot" on page 24 settings (chart output, two-dimensional plot settings, three-dimensional plot settings, and tooltips).

   Note: Plot is available only when **Group sizes** values are specified and **Estimate power** is selected.

Power Analysis of One-way ANOVA: Contrast

You can specify the following contrast, coefficient, and pairwise differences settings for your Power Analysis of One-way ANOVA:

Contrast Test

Test with linear contrasts
When enabled, the contrast and coefficient settings are available.

Test direction

Nondirectional or two-sided analysis
When selected, a two-sided test is used. This is the default setting.

Directional or one-sided analysis
When selected, power is computed for a one-sided test.

Coefficients
Use the table to specify the contrast coefficients and request the contrast test. The table values are optional. The number of specified values must be equal to the values specified for **Group sizes** and **Group means**. The sum for all specified values must equal 0, otherwise the last value will automatically be adjusted.

Pairwise Differences

Estimate the power of testing for pairwise differences
Controls whether or not to estimate the power of testing for the pairwise differences. Be default, the optional setting is disabled, which suppresses output for the pairwise differences.

Adjust the significance level by
Determines the adjustment of multiple comparisons.

Bonferroni correction
Uses the Bonferroni correction in estimating the power of pairwise differences. This is the default setting.

Sidak correction
Uses the Sidak correction in estimating the power of pairwise differences.

Least significant difference (LSD)
Uses the LSD correction in estimating the power of pairwise differences.

Power Analysis of One-way ANOVA: Plot

You can control the plots that are output to illustrate the two and three-dimensional power by sample and effect size charts. You can also control the display of tool tips and the vertical/horizontal rotation degrees for three-dimensional charts.

Two-Dimensional Plot

Power estimation versus total sample size
When enabled, this optional setting provides options for controlling the two-dimensional power by total sample size chart. The setting is disabled by default.

Range of total sample size
When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range of the total sample size is used.

Lower bound
Controls the lower bound for the two-dimensional power by total sample size chart. The value must be greater than, or equal to:

- 2 x the number of integers specified for **Group sizes**
- 2 x the sum of the integers specified for **Group sizes** / by the smallest integer value for **Group sizes**

The value cannot be greater than the **Upper bound** value.

Upper bound

Controls the upper bound for the two-dimensional power by total sample size chart. The value must be less than, or equal to:

- 5000 / by the largest integer value specified for **Group sizes** x the sum of the integers specified for **Group sizes**

The value must be greater than the **Lower bound** value and cannot be greater than 2147483647.

Power estimation versus pooled standard deviation

By default, this optional setting is disabled. The setting controls the two-dimensional power by pooled standard deviation chart. When enabled, the chart displays in the output. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range of the pooled standard deviation is used.

Range of pooled standard deviation

When selected, the lower and upper bound options are available.

Lower bound

Controls the lower bound for the two-dimensional power by pooled standard deviation chart. The value must be greater than 0 and cannot be greater than the **Upper bound** value.

Upper bound

Controls the upper bound for the two-dimensional power by pooled standard deviation chart. The value must be greater than the **Lower bound** value.

Three-Dimensional Plot

Power estimation versus

Provides options for controlling the three-dimensional power by total sample size (x-axis) and effect size (y-axis) chart, the vertical/horizontal rotation settings, and the user specified plot range of sample/effect size. This setting is disabled by default.

Pooled standard deviation on x-axis and total sample size on y-axis

The optional setting controls the three-dimensional power by total sample size (x-axis) and pooled standard deviation (y-axis) chart. By default, the chart is suppressed. When specified, the chart displays.

Pooled standard deviation on y-axis and total sample size on x-axis

The optional setting controls the three-dimensional power by total sample size (y-axis) and pooled standard deviation (x-axis) chart. By default, the chart is suppressed. When specified, the chart displays.

Vertical rotation

The optional setting sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. You can use the mouse to rotate the chart vertically. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 10.

Horizontal rotation

The optional setting sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. You can use the mouse to rotate the chart horizontally. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 325.

Range of total sample size

When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range of the sample size is used.

Lower bound

Controls the lower bound for the three-dimensional power by total sample size chart. The value must be greater than 0 and cannot be greater than the **Upper bound** value.

Upper bound

Controls the upper bound for the three-dimensional power by total sample size chart. The value must be greater than the **Lower bound** value.

Range of pooled standard deviation

When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range of the effect size is used.

Lower bound

Controls the lower bound for the three-dimensional power by pooled standard deviation chart. The value must be greater than 0 and cannot be greater than the **Upper bound** value.

Upper bound

Controls the upper bound for the three-dimensional power by pooled standard deviation chart. The value must be greater than the **Lower bound** value.

Proportions

The following statistics features are included in IBM SPSS Statistics Base Edition.

Power Analysis of Related-Sample Binomial Test

This feature requires IBM SPSS Statistics Base Edition.

Power analysis plays a pivotal role in a study plan, design, and conduction. The calculation of power is usually before any sample data have been collected, except possibly from a small pilot study. The precise estimation of the power may tell investigators how likely it is that a statistically significant difference will be detected based on a finite sample size under a true alternative hypothesis. If the power is too low, there is little chance of detecting a significant difference, and non-significant results are likely even if real differences truly exist.

Binomial distribution is based on a sequence of Bernoulli trials. It can be used to model experiments, including a fixed number of total trials, which are assumed to be independent of each other. Each trial leads to a dichotomous result, with the same probability for a successful outcome.

The related-sample binomial estimates the power of McNemar's test to compare two proportion parameters based on the matched pair subjects sampled from two related binomial populations.

1. From the menus choose:

   Analyze > **Power Analysis** > **Proportions** > **Related-Sample Binomial Test**
2. Select a test assumption setting (**Estimate sample size** or **Estimate power**).
3. When selecting **Estimate power**, enter the appropriate **Total number of pairs** value. The value must be a positive integer greater than, or equal to, 2. When selecting **Estimate sample size**, enter an appropriate **Power** for sample size estimation value. The value must be a single value between 0 and 1.
4. Select to specify testing values for either **Proportions** or **Counts**.
   - When **Proportions** is selected, enter values in the **Proportion 1** and **Proportion 2** fields. The values must be between 0 and 1.
   - When **Counts** is selected, enter values in the **Count 1** and **Count 2** fields. The values must be between 0 and value specified for **Total number of pairs**.

   Proportions Notes:

   - **Proportions** is the only available option when a **Power** value is specified.
   - When **Test values are marginal** is not selected: 0 < **Proportion 1** + **Proportion 2** ≤ 1
   - When **Test values are marginal** is selected:
     - **Proportion 1** * **Proportion 2** > 0

    - **Proportion 1** < 1
    - **Proportion 2** < 1
    - The values for **Proportion 1** and **Proportion 2** cannot be the same.

   Counts Notes:

   - When **Test values are marginal** is not selected: 0 < **Count 1** + **Count 2** ≤ **Total number of pairs**
   - When **Test values are marginal** is selected:
     - **Count 1** * **Count 2** > 0
     - **Count 1** < **Total number of pairs**
     - **Count 2** < **Total number of pairs**

5. You can optionally select **Test values are marginal** to control whether or not the specified proportions or counts values are marginal. When **Test values are marginal** is enabled, you must specify a **Correlation between matched pairs** value. The value must be a single value between -1 and 1.
6. Select a method for estimating the power.

   Normal approximation
   Enables normal approximation. This is the default setting.

   Binomial enumeration
   Enables the binomial enumeration method.
7. Select whether the test is one or two-sided.

   Nondirectional or two-sided analysis
   When selected, a two-sided test is used. This is the default setting.

   Directional or one-sided analysis
   When selected, power is computed for a one-sided test.
8. Optionally, specify the significance level of the Type I error rate for the test in the **Significance level** field. The value must be a single double value between 0 and 1. The default value is 0.05.
9. You can optionally click **Plot** to specify "Power Analysis of Related-Sample Binomial Test" on page 26 settings (chart output, two-dimensional plot settings, three-dimensional plot settings, and tooltips).

   Note: Plot is available only when **Estimate power** is selected as the test assumption.

Power Analysis of Related-Sample Binomial: Plot

You can control the plots that are output to illustrate the two and three-dimensional power estimation charts. You can also control the display of tool tips and the vertical/horizontal rotation degrees for three-dimensional charts.

Two-Dimensional Plot
Provides options for controlling the two-dimensional power estimation versus charts. This setting is disabled by default.

Power estimation versus total number of pairs
When enabled, this optional setting controls the two-dimensional power by total number of pairs chart. The setting is disabled by default. When selected, this setting displays the chart.

Plot range of total number of pairs
When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range is used.

Lower bound
Controls the lower bound for the two-dimensional power estimation versus total number of pairs chart. The value must be greater than 1, and cannot be greater than the **Upper bound** value.

Upper bound

Controls the upper bound for the two-dimensional power estimation versus total number of pairs chart. The value must be greater than the **Lower bound** value and cannot be greater than 2500.

Power estimation versus risk difference

When enabled, this optional setting controls the two-dimensional power by risk difference chart. The setting is disabled by default. When selected, this setting displays the chart.

Power estimation versus risk ratio

When enabled, this optional setting control the two-dimensional power by risk ratio chart. The setting is disabled by default.

Plot range of risk ratio

When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range is used.

Lower bound

Controls the lower bound for the two-dimensional power estimation versus risk ratio chart. The value cannot be greater than the **Upper bound** value.

Upper bound

Controls the upper bound for the two-dimensional power estimation versus risk ratio chart. The value must be greater than the **Lower bound** value and cannot be greater than 10.

Power estimation versus odds ratio

When enabled, this optional setting controls the two-dimensional power by odds ratio chart. The setting is disabled by default. When selected, this setting displays the chart.

Plot range of odds ratio

When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range is used.

Lower bound

Controls the lower bound for the two-dimensional power estimation versus odds ratio chart. The value cannot be greater than the **Upper bound** value.

Upper bound

Controls the upper bound for the two-dimensional power estimation versus odds ratio chart. The value must be greater than the **Lower bound** value and cannot be greater than 10.

Three-Dimensional Plot

Provides options for controlling the three-dimensional power estimation versus charts. This setting is disabled by default.

Power estimation versus discordant proportions

When enabled, this optional setting controls the three-dimensional power by discordant proportions chart. The setting is disabled by default. When selected, this setting displays the chart.

Power estimation versus marginal proportions

When enabled, this optional setting controls the three-dimensional power by marginal proportions chart. The setting is disabled by default. When selected, this setting displays the chart.

Note: This setting is available only when **Test values are marginal** is selected.

Vertical rotation

The optional setting sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. You can use the mouse to rotate the chart vertically. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 10.

Horizontal rotation

The optional setting sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. You can use the mouse to rotate the chart horizontally. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 325.

Power Analysis of Independent-Sample Binomial Test

This feature requires IBM SPSS Statistics Base Edition.

Power analysis plays a pivotal role in a study plan, design, and conduction. The calculation of power is usually before any sample data have been collected, except possibly from a small pilot study. The precise estimation of the power may tell investigators how likely it is that a statistically significant difference will be detected based on a finite sample size under a true alternative hypothesis. If the power is too low, there is little chance of detecting a significant difference, and non-significant results are likely even if real differences truly exist.

The binomial distribution is based on a sequence of Bernoulli trials. It can be used to model those experiments including a fixed number of total trials that are assumed to be independent of each other. Each trial leads to a dichotomous result, with the same probability for a "successful" outcome. The independent-sample binomial test compares two independent proportion parameters.

1. From the menus choose:

   Analyze > **Power Analysis** > **Proportions** > **Independent-Samples Binomial Test**
2. Select a test assumption setting (**Estimate sample size** or **Estimate power**).
3. When **Estimate sample size** is selected, enter an appropriate **Power** for sample size estimation value (the value must be a single value between 0 and 1) and a **Group size ratio** value for specifying the ratio of the sample sizes (the value must be a single value between 0.01 and 100.
4. When **Estimate power** is selected, enter values to specify the total number of trials for both **group 1** and **group 2**. The values must be an integers greater than 1.
5. Specify the proportion parameters for the two groups. Both values must be between 0 and 1.

   Note: The two values cannot be the same when a **Power** value is specified.
6. Optionally, specify the significance level of the Type I error rate for the test in the **Significance level** field. The value must be a single double value between 0 and 1. The default value is 0.05.
7. Select the desired test assumptions:

   Chi-squared test
   Estimates the power based on Pearson's chi-squared test. This is the default setting.

   Standard deviation is pooled
   This optional setting controls whether the estimation of the standard deviation is pooled or unpooled. The setting is enabled by default.

   Apply continuity correction
   This optional setting controls whether or not the continuity correction is used. The setting is disabled by default.

   T-test
   Estimates the power based on Student's t-test.

   Standard deviation is pooled
   This optional setting controls whether the estimation of the standard deviation is pooled or unpooled. The setting is enabled by default.

   Likelihood ratio test
   Estimates the power based on the likelihood ratio test.

   Fisher's exact test
   Estimates the power based on Fisher's exact test.

   Note: In some cases, Fisher's exact test may take an extended amount of time to complete.
8. Select a method for estimating the power.

   Normal approximation
   Enables normal approximation. This is the default setting.

Binomial enumeration
Enables the binomial enumeration method.

9. Select whether the test is one or two-sided.

   Nondirectional or two-sided analysis
   When selected, a two-sided test is used. This is the default setting.

   Directional or one-sided analysis
   When selected, power is computed for a one-sided test.

10. You can optionally click **Plot** to specify "Power Analysis of Independent-Samples Binomial Test: Plot" on page 30 settings (chart output, two-dimensional plot settings, three-dimensional plot settings, and tooltips).

    Note: Plot is available only when **Estimate power** is selected as the test assumption.

Power Analysis of Independent-Samples Binomial Test: Plot

You can control the plots that are output to illustrate the two and three-dimensional power estimation charts. You can also control the display of tool tips and the vertical/horizontal rotation degrees for three-dimensional charts.

Two-Dimensional Plot
Provides options for controlling the two-dimensional power estimation versus charts. This setting is disabled by default.

Power estimation versus group size ratio
When enabled, this optional setting controls the two-dimensional power by group size ratio chart. The setting is disabled by default. When selected, this setting displays the chart.

Plot range of group size ratio
When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range is used.

Lower bound
Controls the lower bound for the two-dimensional power estimation versus total number of pairs chart. The value must be greater than .01, and cannot be greater than the **Upper bound** value.

Upper bound
Controls the upper bound for the two-dimensional power estimation versus total number of pairs chart. The value must be greater than the **Lower bound** value and cannot be greater than 100.

Power estimation versus risk difference
When enabled, this optional setting controls the two-dimensional power by risk difference chart. The setting is disabled by default. When selected, this setting displays the chart.

Power estimation versus risk ratio
When enabled, this optional setting control the two-dimensional power by risk ratio chart. The setting is disabled by default.

Plot range of risk ratio
When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range is used.

Lower bound
Controls the lower bound for the two-dimensional power estimation versus risk ratio chart. The value cannot be greater than the **Upper bound** value.

Upper bound
Controls the upper bound for the two-dimensional power estimation versus risk ratio chart. The value must be greater than the **Lower bound** value and cannot be greater than 10.

Power estimation versus odds ratio

When enabled, this optional setting controls the two-dimensional power by odds ratio chart. The setting is disabled by default. When selected, this setting displays the chart.

Plot range of odds ratio

When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range is used.

Lower bound

Controls the lower bound for the two-dimensional power estimation versus odds ratio chart. The value cannot be greater than the **Upper bound** value.

Upper bound

Controls the upper bound for the two-dimensional power estimation versus odds ratio chart. The value must be greater than the **Lower bound** value and cannot be greater than 10.

Three-Dimensional Plot

Provides options for controlling the three-dimensional power estimation versus charts. This setting is disabled by default.

Power estimation versus proportions

When selected, this optional setting provides the following power by proportion options:

proportion of group 1 on x-axis and proportion of group 2 on y-axis

Controls the three-dimensional power by proportion of Group 1 (x -axis) and proportion of Group 2 (y-axis) chart. The setting is disabled by default. When selected, this setting displays the chart.

proportion of group 1 on y-axis and proportion of group 2 on x-axis

Controls the three-dimensional power by proportion of Group 2 (x -axis) and proportion of Group 1 (y-axis) chart. The setting is disabled by default. When selected, this setting displays the chart.

Power estimation versus group sizes

When selected, this optional setting provides the following power by group sizes options:

size of group 1 on x-axis and size of group 2 on y-axis

Controls the three-dimensional power by number of trials in Group 1 (x -axis) and number of trials in Group 2 (y-axis) chart. The setting is disabled by default. When selected, this setting displays the chart.

size of group 1 on y-axis and size of group 2 on x-axis

Controls the three-dimensional power by number of trials in Group 2 (x -axis) and number of trials in Group 1 (y-axis) chart. The setting is disabled by default. When selected, this setting displays the chart.

User specified plot range of size of group 1

When selected, the lower and upper bound options for the group 1 plot range are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range is used.

Lower bound

Controls the lower bound for the two-dimensional power estimation versus odds ratio chart. The value must be greater than or equal to 2 and cannot be greater than the **Upper bound** value.

Upper bound

Controls the upper bound for the two-dimensional power estimation versus odds ratio chart. The value must be greater than the **Lower bound** value and cannot be greater than 2500.

User specified plot range of size of group 2

When selected, the lower and upper bound options for the group 2 plot range are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range is used.

Lower bound

Controls the lower bound for the two-dimensional power estimation versus odds ratio chart. The value must be greater than or equal to 2 and cannot be greater than the **Upper bound** value.

Upper bound

Controls the upper bound for the two-dimensional power estimation versus odds ratio chart. The value must be greater than the **Lower bound** value and cannot be greater than 2500.

Vertical rotation

The optional setting sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. You can use the mouse to rotate the chart vertically. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 10.

Horizontal rotation

The optional setting sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. You can use the mouse to rotate the chart horizontally. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 325.

Power Analysis of One-Sample Binomial Test

This feature requires IBM SPSS Statistics Base Edition.

Power analysis plays a pivotal role in a study plan, design, and conduction. The calculation of power is usually before any sample data have been collected, except possibly from a small pilot study. The precise estimation of the power may tell investigators how likely it is that a statistically significant difference will be detected based on a finite sample size under a true alternative hypothesis. If the power is too low, there is little chance of detecting a significant difference, and non-significant results are likely even if real differences truly exist.

Binomial distribution is based on a sequence of Bernoulli trials. It can be used to model experiments, including a fixed number of total trials, which are assumed to be independent of each other. Each trial leads to a dichotomous result, with the same probability for a successful outcome.

The one-sample binomial test makes statistical inference about the proportion parameter by comparing it with a hypothesized value. The methods for estimating the power for such a test are either the normal approximation or the binomial enumeration.

1. From the menus choose:

   Analyze > **Power Analysis** > **Proportions** > **One-Sample Binomial Test**
2. Select a test assumption setting (**Estimate sample size** or **Estimate power**).
3. When selecting **Estimate power**, enter the appropriate **Total number of trials** value. The value must be an integer greater than, or equal to, 1. When selecting **Estimate sample size**, enter an appropriate **Power** for sample size estimation value. The value must be a single value between 0 and 1.
4. Enter a value that specifies the alternative hypothesis value of the proportion parameter in the **Population proportion** field. The value must be a single numeric.

   Note: When a **Power** value is specified, the **Population proportion**value cannot be equal to the **Null value**.
5. Optionally, enter a value that specifies the null hypothesis value of the proportion parameter to be tested in the **Null value** field. The value must be a single numeric between 0 and 1. The default value is 0.50.
6. Select a method for estimating the power.

   Normal approximation

   Enables normal approximation. This is the default setting.

   Apply continuity correction

   Control whether or not the continuity correction is used for the normal approximation method.

Binomial enumeration
Enables the binomial enumeration method.

Note: The selected power estimation method has no effect when the **Total number of trials** value exceeds 500.

7. Select whether the test is one or two-sided.

   Nondirectional or two-sided analysis
   When selected, a two-sided test is used. This is the default setting.

   Directional or one-sided analysis
   When selected, power is computed for a one-sided test.

8. Optionally, specify the significance level of the Type I error rate for the test in the **Significance level** field. The value must be a single double value between 0 and 1. The default value is 0.05.
9. You can optionally click **Plot** to specify "Power Analysis of One-Sample Binomial: Plot" on page 33 settings (chart output, two-dimensional plot settings, three-dimensional plot settings, and tooltips).

   Note: Plot is available only when **Estimate power** is selected as the test assumption.

Power Analysis of One-Sample Binomial: Plot

You can control the plots that are output to illustrate the two and three-dimensional power by total number of trials charts. You can also control the display of tool tips and the vertical/horizontal rotation degrees for three-dimensional charts.

Two-Dimensional Plot
Provides options for controlling the two-dimensional power estimation versus charts. This setting is disabled by default.

Power estimation versus null hypothesis value
When enabled, this optional setting controls the two-dimensional power by null value chart. The setting is disabled by default. When selected, this setting displays the chart.

Power estimation versus alternative hypothesis value
When enabled, this optional setting controls the two-dimensional power by alternative value chart. The setting is disabled by default. When selected, this setting displays the chart.

Power estimation versus the difference between hypothesized values
When enabled, this optional setting control the two-dimensional power by risk difference chart. The setting is disabled by default.

Power estimation versus total number of trials
When enabled, this optional setting controls the two-dimensional power by total number of trials chart. The setting is disabled by default. When selected, this setting displays the chart.

Plot range of total number of trials
When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range is used.

Lower bound
Controls the lower bound for the two-dimensional power estimation versus total number of trials chart. The value must be greater than 0, and cannot be greater than the **Upper bound** value.

Upper bound
Controls the upper bound for the two-dimensional power estimation versus total number of trials chart. The value must be greater than the **Lower bound** value and cannot be greater than 5000.

Three-Dimensional Plot
Provides options for controlling the three-dimensional power estimation versus charts. This setting is disabled by default.

Power estimation versus total number of trials
When selected, this setting enables the following options.

on x-axis and the difference between hypothesized values on y-axis
The optional setting controls the three-dimensional power by total number of trials (x -axis) and risk difference (y-axis) chart. By default, the chart is suppressed. When specified, the chart displays.

on y-axis and the difference between hypothesized values on x-axis
The optional setting controls the three-dimensional power by total number of trials (y-axis) and risk difference (x -axis) chart. By default, the chart is suppressed. When specified, the chart displays.

Plot range of total number of trials
When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range is used.

Lower bound
Controls the lower bound for the three-dimensional power estimation versus total number of trials chart. The value must be greater than 0, and cannot be greater than the **Upper bound** value.

Upper bound
Controls the upper bound for the three-dimensional power estimation versus total number of trials chart. The value must be greater than the **Lower bound** value and cannot be greater than 5000.

Power estimation versus null hypothesis value
When selected, this setting enables the following options.

on x-axis and alternative hypothesis value on y-axis
The optional setting controls the three-dimensional power by null (x -axis) and alternative value (y-axis) chart. By default, the chart is suppressed. When specified, the chart displays.

on y-axis and alternative hypothesis value on x-axis
The optional setting controls the three-dimensional power by null (y-axis) and alternative value (x -axis) chart.. By default, the chart is suppressed. When specified, the chart displays.

Vertical rotation
The optional setting sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. You can use the mouse to rotate the chart vertically. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 10.

Horizontal rotation
The optional setting sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. You can use the mouse to rotate the chart horizontally. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 325.

Correlations

The following statistics features are included in IBM SPSS Statistics Base Edition.

Power Analysis of One-Sample Pearson Correlation Test

This feature requires IBM SPSS Statistics Base Edition.

Power analysis plays a pivotal role in a study plan, design, and conduction. The calculation of power is usually before any sample data have been collected, except possibly from a small pilot study. The precise estimation of the power may tell investigators how likely it is that a statistically significant difference will be detected based on a finite sample size under a true alternative hypothesis. If the power is too low, there is little chance of detecting a significant difference, and non-significant results are likely even if real differences truly exist.

Pearson's product-moment correlation coefficient measures the strength of linear association between two scale random variables that are assumed to follow a bivariate normal distribution. By convention, it is a dimensionless quantity and obtained by standardizing the covariance between two continuous variables, thereby ranging between -1 and 1.

The test uses Fisher's asymptotic method to estimate the power for the one-sample Pearson correlation.

1. From the menus choose:

   Analyze > **Power Analysis** > **Correlations** > **Pearson Product-Moment**
2. Select a test assumption setting (**Estimate sample size** or **Estimate power**).
3. When selecting **Estimate power**, enter the appropriate **Sample size in pairs** value. The value must be a single integer greater than 3. When selecting **Estimate sample size**, enter a **Power** for sample size estimation value. The value must be a single value between 0 and 1.
4. Enter a value that specifies the alternative hypothesis value of the correlation parameter in the **Pearson correlation parameter** field. The value must be a single numeric between -1 and 1.

   Note: When a **Power** value is specified, the **Pearson correlation parameter** value cannot be -1 or 1 and cannot be equal **Null value**.
5. Optionally, enter a value that specifies the null hypothesis value of the correlation parameter to be tested in the **Null value** field. The value must be a single numeric between -1 and 1. The default value is 0.

   Note: When a **Power** value is specified, **Null value** cannot be -1 or 1.
6. Optionally, select **Use bias-correction formiula in the power estimation** to specify whether the bias adjustment is involved or ignored. The setting is enabled by default, which includes the bias adjustment term in the power estimation. When the setting is not selected, the bias adjustment term is ignored.
7. Select whether the test is one or two-sided.

   Nondirectional or two-sided analysis
   When selected, a two-sided test is used. This is the default setting.

   Directional or one-sided analysis
   When selected, power is computed for a one-sided test.
8. Optionally, specify the significance level of the Type I error rate for the test in the **Significance level** field. The value must be a single double value between 0 and 1. The default value is 0.05.
9. You can optionally click **Plot** to specify "Power Analysis of One-Sample Pearson Correlation: Plot" on page 35 settings (chart output, two-dimensional plot settings, three-dimensional plot settings, and tooltips).

   Note: Plot is available only when **Estimate power** is selected as the test assumption.

Power Analysis of One-Sample Pearson Correlation: Plot

You can control the plots that are output to illustrate the two and three-dimensional power by charts. You can also control the display of tool tips and the vertical/horizontal rotation degrees for three-dimensional charts.

Two-Dimensional Plot
Provides options for controlling the two-dimensional power estimation versus charts. The settings are disabled by default.

Power estimation versus null hypothesis value
When enabled, this optional setting controls the two-dimensional power by null value chart. The setting is disabled by default. When selected, this setting displays the chart.

Power estimation versus alternative hypothesis value
When enabled, this optional setting controls the two-dimensional power by alternative value chart. The setting is disabled by default. When selected, this setting displays the chart.

Power estimation versus the difference between hypothesized values
When enabled, this optional setting control the two-dimensional power by risk difference chart. The setting is disabled by default.

Power estimation versus sample size (in pairs)
When enabled, this optional setting controls the two-dimensional power by sample size chart. The setting is disabled by default. When selected, this setting displays the chart.

Plot range of sample size
When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range is used.

Lower bound
Controls the lower bound for the two-dimensional power estimation by sample size chart. The value must be greater than or equal to 4, and cannot be greater than the **Upper bound** value.

Upper bound
Controls the upper bound for the two-dimensional power estimation by sample size chart. The value must be greater than the **Lower bound** value and cannot be greater than 5000.

Three-Dimensional Plot
Provides options for controlling the three-dimensional power estimation versus charts. This setting is disabled by default.

Power estimation versus sample size
When selected, this setting enables the following options.

on x-axis and the difference between hypothesized values on y-axis
The optional setting controls the three-dimensional power by sample size (x -axis) and difference between hypothesized values (y-axis) chart. By default, the chart is suppressed. When specified, the chart displays.

on y-axis and the difference between hypothesized values on x-axis
The optional setting controls the three-dimensional power by sample size (y -axis) and difference between hypothesized values (x-axis) chart. By default, the chart is suppressed. When specified, the chart displays

Plot range of sample size (in pairs)
When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range is used.

Lower bound
Controls the lower bound for the three-dimensional power estimation by sample size chart. The value must be greater than or equal to 4, and cannot be greater than the **Upper bound** value.

Upper bound
Controls the upper bound for the three-dimensional power estimation by sample size chart. The value must be greater than the **Lower bound** value and cannot be greater than 5000.

Power estimation versus null hypothesis value
When selected, this setting enables the following options.

on x-axis and alternative hypothesis value on y-axis
The optional setting controls the three-dimensional power by null (x -axis) and alternative value (y-axis) chart. By default, the chart is suppressed. When specified, the chart displays.

on y-axis and alternative hypothesis value on x-axis
The optional setting controls the three-dimensional power by null (y-axis) and alternative value (x -axis) chart.. By default, the chart is suppressed. When specified, the chart displays.

Vertical rotation
The optional setting sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. You can use the mouse to rotate the chart vertically. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 10.

Horizontal rotation
The optional setting sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. You can use the mouse to rotate the chart horizontally. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 325.

Power Analysis of One-Sample Spearman Correlation Test

This feature requires IBM SPSS Statistics Base Edition.

Power analysis plays a pivotal role in a study plan, design, and conduction. The calculation of power is usually before any sample data have been collected, except possibly from a small pilot study. The precise estimation of the power may tell investigators how likely it is that a statistically significant difference will be detected based on a finite sample size under a true alternative hypothesis. If the power is too low, there is little chance of detecting a significant difference, and non-significant results are likely even if real differences truly exist.

Spearman rank-order correlation coefficient is a rank-based nonparametric statistic to measure the monotonic relationship between two variables that are usually censored and not normally distributed. The Spearman rank-order correlation is equal to the Pearson correlation between the rank values of the two variables, thereby also ranging between -1 and 1. Detecting the power of the Spearman rank correlation test is an important topic in the analysis of hydrological time series data.

The test uses Fisher's asymptotic method to estimate the power for the one-sample Spearman rank-order correlation.

1. From the menus choose:

   Analyze > **Power Analysis** > **Correlations** > **Spearman Rank-Order**
2. Select a test assumption setting (**Estimate sample size** or **Estimate power**).
3. When selecting **Estimate power**, enter the appropriate **Sample size in pairs** value. The value must be a single integer greater than 3. When selecting **Estimate sample size**, enter a **Power** for sample size estimation value. The value must be a single value between 0 and 1.
4. Enter a value that specifies the alternative hypothesis value of the correlation parameter in the **Spearman correlation parameter** field. The value must be a single numeric between -1 and 1.

   Note: When a **Power** value is specified, the **Spearman correlation parameter** value cannot be -1 or 1 and cannot be equal **Null value**.
5. Optionally, enter a value that specifies the null hypothesis value of the correlation parameter to be tested in the **Null value** field. The value must be a single numeric between -1 and 1. The default value is 0.

   Note: When a **Power** value is specified, **Null value** cannot be -1 or 1.
6. Optionally, select an option that determines how the asymptotic variance is estimated for the power analysis.

   Bonett and Wright
   Estimates the variance suggested by Bonett and Wright. This is the default setting.

   Fieller, Hartley and Pearson
   Estimates the variance suggested by Fieller, Hartley and Pearson.

   Caruso and Cliff
   Estimates the variance suggested by Caruso and Cliff.
7. Select whether the test is one or two-sided.

   Nondirectional or two-sided analysis
   When selected, a two-sided test is used. This is the default setting.

   Directional or one-sided analysis
   When selected, power is computed for a one-sided test.
8. Optionally, specify the significance level of the Type I error rate for the test in the **Significance level** field. The value must be a single double value between 0 and 1. The default value is 0.05.

9. You can optionally click **Plot** to specify "Power Analysis of One-Sample Spearman Correlation: Plot" on page 38 settings (chart output, two-dimensional plot settings, three-dimensional plot settings, and tooltips).

   Note: Plot is available only when **Estimate power** is selected as the test assumption.

Power Analysis of One-Sample Spearman Correlation: Plot

You can control the plots that are output to illustrate the two and three-dimensional power by charts. You can also control the display of tool tips and the vertical/horizontal rotation degrees for three-dimensional charts.

Two-Dimensional Plot
Provides options for controlling the two-dimensional power estimation versus charts. The settings are disabled by default.

Power estimation versus null hypothesis value
When enabled, this optional setting controls the two-dimensional power by null value chart. The setting is disabled by default. When selected, this setting displays the chart.

Power estimation versus alternative hypothesis value
When enabled, this optional setting controls the two-dimensional power by alternative value chart. The setting is disabled by default. When selected, this setting displays the chart.

Power estimation versus the difference between hypothesized values
When enabled, this optional setting control the two-dimensional power by risk difference chart. The setting is disabled by default.

Power estimation versus sample size (in pairs)
When enabled, this optional setting controls the two-dimensional power by sample size chart. The setting is disabled by default. When selected, this setting displays the chart.

Plot range of sample size
When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range is used.

Lower bound
Controls the lower bound for the two-dimensional power estimation by sample size chart. The value must be greater than or equal to 4, and cannot be greater than the **Upper bound** value.

Upper bound
Controls the upper bound for the two-dimensional power estimation by sample size chart. The value must be greater than the **Lower bound** value and cannot be greater than 5000.

Three-Dimensional Plot
Provides options for controlling the three-dimensional power estimation versus charts. This setting is disabled by default.

Power estimation versus sample size
When selected, this setting enables the following options.

on x-axis and the difference between hypothesized values on y-axis
The optional setting controls the three-dimensional power by sample size (x -axis) and difference between hypothesized values (y-axis) chart. By default, the chart is suppressed. When specified, the chart displays.

on y-axis and the difference between hypothesized values on x-axis
The optional setting controls the three-dimensional power by sample size (y -axis) and difference between hypothesized values (x-axis) chart. By default, the chart is suppressed. When specified, the chart displays

Plot range of sample size (in pairs)
When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range is used.

Lower bound
Controls the lower bound for the three-dimensional power estimation by sample size chart. The value must be greater than or equal to 4, and cannot be greater than the **Upper bound** value.

Upper bound
Controls the upper bound for the three-dimensional power estimation by sample size chart. The value must be greater than the **Lower bound** value and cannot be greater than 5000.

Power estimation versus null hypothesis value
When selected, this setting enables the following options.

on x-axis and alternative hypothesis value on y-axis
The optional setting controls the three-dimensional power by null (x -axis) and alternative value (y-axis) chart. By default, the chart is suppressed. When specified, the chart displays.

on y-axis and alternative hypothesis value on x-axis
The optional setting controls the three-dimensional power by null (y-axis) and alternative value (x -axis) chart.. By default, the chart is suppressed. When specified, the chart displays.

Vertical rotation
The optional setting sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. You can use the mouse to rotate the chart vertically. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 10.

Horizontal rotation
The optional setting sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. You can use the mouse to rotate the chart horizontally. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 325.

Power Analysis of Partial Pearson Correlation Test

This feature requires IBM SPSS Statistics Base Edition.

Power analysis plays a pivotal role in a study plan, design, and conduction. The calculation of power is usually before any sample data have been collected, except possibly from a small pilot study. The precise estimation of the power may tell investigators how likely it is that a statistically significant difference will be detected based on a finite sample size under a true alternative hypothesis. If the power is too low, there is little chance of detecting a significant difference, and non-significant results are likely even if real differences truly exist.

Partial correlation can be explained as the association between two random variables after eliminating the effect of another or several other variables. It is a useful measurement in the presence of confounding. Similar to the Pearson correlation coefficient, partial correlation coefficient is also a dimensionless quantity ranging between -1 and 1.

The test uses Fisher's asymptotic method to estimate the power for the one-sample Pearson correlation.

1. From the menus choose:

   Analyze > **Power Analysis** > **Correlations** > **Partial**
2. Select a test assumption setting (**Estimate sample size** or **Estimate power**).
3. When selecting **Estimate power**, enter the appropriate **Sample size in pairs** value. The value must be a single integer greater than 3. When selecting **Estimate sample size**, enter a **Power** for sample size estimation value. The value must be a single value between 0 and 1.
4. Enter a value that specifies the number of the variables assumed to be partialed out. The value must be a single integer greater than or equal to 0.
5. Enter a value that specifies the alternative hypothesis value of the partial correlation parameter in the **Partial Pearson correlation parameter** field. The value must be a single numeric between -1 and 1.

Note: When a **Power** value is specified, the **Partial Pearson correlation parameter** value cannot be -1 or 1 and cannot be equal **Null value**.

6. Optionally, enter a value that specifies the null hypothesis value of the partial correlation parameter to be tested in the **Null value** field. The value must be a single numeric between -1 and 1. The default value is 0.

   Note: When a **Power** value is specified, **Null value** cannot be -1 or 1.

7. Select whether the test is one or two-sided.

   Nondirectional or two-sided analysis
   When selected, a two-sided test is used. This is the default setting.

   Directional or one-sided analysis
   When selected, power is computed for a one-sided test.

8. Optionally, specify the significance level of the Type I error rate for the test in the **Significance level** field. The value must be a single double value between 0 and 1. The default value is 0.05.

9. You can optionally click **Plot** to specify "Power Analysis of Partial Pearson Correlation: Plot" on page 40 settings (chart output, two-dimensional plot settings, three-dimensional plot settings, and tooltips).

   Note: Plot is available only when **Estimate power** is selected as the test assumption.

Power Analysis of Partial Pearson Correlation: Plot

You can control the plots that are output to illustrate the two and three-dimensional power by charts. You can also control the display of tool tips and the vertical/horizontal rotation degrees for three-dimensional charts.

Two-Dimensional Plot
Provides options for controlling the two-dimensional power estimation versus charts. The settings are disabled by default.

Power estimation versus null hypothesis value
When enabled, this optional setting controls the two-dimensional power by null value chart. The setting is disabled by default. When selected, this setting displays the chart.

Power estimation versus alternative hypothesis value
When enabled, this optional setting controls the two-dimensional power by alternative value chart. The setting is disabled by default. When selected, this setting displays the chart.

Power estimation versus the number of variables partialed out
When enabled, this optional setting controls the two-dimensional power by number of partialed-out variables chart. The setting is disabled by default. When selected, this setting displays the chart.

Power estimation versus the difference between hypothesized values
When enabled, this optional setting control the two-dimensional power by risk difference chart. The setting is disabled by default.

Power estimation versus sample size
When enabled, this optional setting controls the two-dimensional power by sample size chart. The setting is disabled by default. When selected, this setting displays the chart.

Plot range of sample size
When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range is used.

Lower bound
Controls the lower bound for the two-dimensional power estimation by sample size chart. The value must be greater than or equal to 4, and cannot be greater than the **Upper bound** value.

Upper bound
Controls the upper bound for the two-dimensional power estimation by sample size chart. The value must be greater than the **Lower bound** value and cannot be greater than 5000.

Three-Dimensional Plot

Provides options for controlling the three-dimensional power estimation versus charts. This setting is disabled by default.

Power estimation versus sample size

When selected, this setting enables the following options.

on x-axis and the difference between hypothesized values on y-axis

The optional setting controls the three-dimensional power by sample size (x -axis) and difference between hypothesized values (y-axis) chart. By default, the chart is suppressed. When specified, the chart displays.

on y-axis and the difference between hypothesized values on x-axis

The optional setting controls the three-dimensional power by sample size (y -axis) and difference between hypothesized values (x-axis) chart. By default, the chart is suppressed. When specified, the chart displays

Plot range of sample size (in pairs)

When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range is used.

Lower bound

Controls the lower bound for the three-dimensional power estimation by sample size chart. The value must be greater than or equal to 4, and cannot be greater than the **Upper bound** value.

Upper bound

Controls the upper bound for the three-dimensional power estimation by sample size chart. The value must be greater than the **Lower bound** value and cannot be greater than 5000.

Power estimation versus null hypothesis value

When selected, this setting enables the following options.

on x-axis and alternative hypothesis value on y-axis

The optional setting controls the three-dimensional power by null (x -axis) and alternative value (y-axis) chart. By default, the chart is suppressed. When specified, the chart displays.

on y-axis and alternative hypothesis value on x-axis

The optional setting controls the three-dimensional power by null (y-axis) and alternative value (x -axis) chart.. By default, the chart is suppressed. When specified, the chart displays.

Vertical rotation

The optional setting sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. You can use the mouse to rotate the chart vertically. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 10.

Horizontal rotation

The optional setting sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. You can use the mouse to rotate the chart horizontally. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 325.

Regression

The following statistics features are included in IBM SPSS Statistics Base Edition.

Power Analysis of Univariate Linear Regression Test

This feature requires IBM SPSS Statistics Base Edition.

Power analysis plays a pivotal role in a study plan, design, and conduction. The calculation of power is usually before any sample data have been collected, except possibly from a small pilot study. The precise estimation of the power may tell investigators how likely it is that a statistically significant difference will be detected based on a finite sample size under a true alternative hypothesis. If the power is too low, there is little chance of detecting a significant difference, and non-significant results are likely even if real differences truly exist.

Univariate linear regression is a basic and standard statistical approach in which researchers use the values of several variables to explain or predict values of a scale outcome.

The Power Analysis of Univariate Linear Regression test estimates the power of the type III *F*-test in univariate multiple linear regression models. With the effect size represented by multiple (partial) correlations, approaches for both fixed and random predictors are provided. For fixed predictors, the power estimation is based on the non-central *F*-distribution. For random predictors, it is assumed that the target variable and the predictors jointly follow a multivariate normal distribution. In which case, power estimation is based on the distribution of the sample multiple correlation coefficient.

1. From the menus choose:

   Analyze > **Power Analysis** > **Regression** > **Univariate Linear**
2. Select a test assumption setting (**Estimate sample size** or **Estimate power**).
3. When selecting **Estimate power**, enter an appropriate **Sample size** for power estimation value. The value must be a single integer greater than or equal to the total number of model predictors +2 (when **Include the intercept term in the model** is enabled). Otherwise the value must be a single integer greater than or equal to the total number of model predictors +1.

   When selecting **Estimate sample size**, enter an appropriate **Power** for sample size estimation value. The value must be a single value between 0 and 1.
4. Specify the value of the multiple partial correlation coefficient in the **Population multiple partial correlation** field. The value must be a single value between -1 and 1.

   Note: When a **Power** value is specified, the **Population multiple partial correlation** value cannot be 0.

   The following settings are enabled when **Population multiple partial correlation** is selected:

   Total number of predictors in the model
   Specify the number of either the total predictors, or the predictors in the full model (not including the intercept, if applicable). The value must be a single integer greater than or equal to 1.

   Number of test predictors
   Specify the number of either the test predictors, or the predictors in the nested model (not including the intercept, if applicable). The value must be greater than or equal to 1, but no larger than the **Total number of predictors in the model** value.
5. Specify **R-squared values for** multiple correlation coefficients for both **Full model** and **Nested model**. The values must be a single values between 0 and 1.

   Note: When a **Power** value is specified, the **Full model** value must be greater than the **Nested model** value.

   The following settings are enabled when **R-squared values for** is selected:

   Total number of predictors - Full Model
   Specify the number of either the total predictors, or the predictors in the full model (not including the intercept, if applicable). The value must be a single integer greater than or equal to 1.

   Total number of predictors - Nested Model
   Specify the number of either the total predictors, or the predictors in the nested model (not including the intercept, if applicable). The value must be greater than or equal to 1, but less than the **Total number of predictors in the model** value.
6. Optionally, specify the significance level of the Type I error rate for the test in the **Significance level** field. The value must be a single double value between 0 and 1. The default value is 0.05.
7. You can optionally select the **Include the intercept term in the model** setting. The setting is enabled by default. When not selected, the intercept term is excluded from the power analysis.
8. You can optionally select whether model predictors are **Fixed** or **Random**. **Fixed** is the default setting.
9. You can optionally click **Plot** to specify "Power Analysis of Univariate Linear Regression: Plot" on page 43 settings (chart output, two-dimensional plot settings, and three-dimensional plot settings).

Note: **Plot** is available only when **Estimate power** is selected as the test assumption.

Power Analysis of Univariate Linear Regression: Plot

You can control the plots that are output to illustrate the two and three-dimensional power by charts. You can also control the display of tool tips and the vertical/horizontal rotation degrees for three-dimensional charts.

Two-Dimensional Plot

Provides options for controlling the two-dimensional power estimation versus charts. The settings are disabled by default.

Power estimation versus the multiple partial correlation

When enables, this optional setting controls the two-dimensional power by multiple partial correlation coefficient chart. The setting is disabled by default. When selected, this setting displays the chart.

Power estimation versus sample size

When enabled, this optional setting controls the two-dimensional power by sample size chart. The setting is disabled by default. When selected, this setting displays the chart.

Plot range of sample size

When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range is used.

Lower bound

Controls the lower bound for the two-dimensional power estimation by sample size chart. The value must be greater than or equal to 4, and cannot be greater than the **Upper bound** value.

Upper bound

Controls the upper bound for the two-dimensional power estimation by sample size chart. The value must be greater than the **Lower bound** value and cannot be greater than 5000.

Three-Dimensional Plot

Provides options for controlling the three-dimensional power estimation versus charts, the vertical/horizontal rotation settings, and the user specified plot range of sample size. This setting is disabled by default.

Power estimation versus sample size

When enabled, this optional setting controls the three-dimensional power by sample size charts. The setting is disabled by default. When selected, this setting displays the chart.

on x-axis and the multiple partial correlation on y-axis

This optional setting controls the three-dimensional power by sample size (x-axis) and multiple partial correlation coefficient (y-axis) chart. The setting is disabled by default. When selected, this setting displays the chart.

on y-axis and the multiple partial correlation on x-axis

This optional setting controls the three-dimensional power by sample size (y-axis) and multiple partial correlation coefficient (x-axis) chart. The setting is disabled by default. When selected, this setting displays the chart.

Plot range of sample size

When selected, the lower and upper bound options are available. When no integer values are specified for the **Lower bound** or **Upper bound** fields, the default plot range is used.

Lower bound

Controls the lower bound for the two-dimensional power estimation by sample size chart. The value must be greater than or equal to 4, and cannot be greater than the **Upper bound** value.

Upper bound

Controls the upper bound for the two-dimensional power estimation by sample size chart. The value must be greater than the **Lower bound** value and cannot be greater than 5000.

Vertical rotation

The optional setting sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. You can use the mouse to rotate the chart vertically. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 10.

Horizontal rotation

The optional setting sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. You can use the mouse to rotate the chart horizontally. The setting takes effect when the three-dimensional plot is requested. The value must be a single integer value less than or equal to 359. The default value is 325.

Chapter 3. Command enhancements

CORRELATIONS

`CORRELATIONS` is available in Statistics Base Edition.

```
CORRELATIONS VARIABLES= varlist [WITH varlist] [/varlist...]

 [/MISSING={PAIRWISE**}  [{INCLUDE}]]
           {LISTWISE  }   {EXCLUDE}

 [/PRINT={TWOTAIL**}  {SIG**} {FULL**}]
         {ONETAIL  }  {NOSIG} {LOWER}

 [/MATRIX=OUT({*                   })]
             {'savfile'|'dataset'}

 [/STATISTICS=[DESCRIPTIVES] [XPROD] [ALL]]
```

**Default if the subcommand is omitted.

This command reads the active dataset and causes execution of any pending commands. See the topic Command Order for more information.

Release History

Release 27.0

- Added support for the **FULL** and **LOWER** keywords in the **PRINT** subcommand.

Release 16.0

- Added support for `SET THREADS` and `SET MCACHE`.

Example

```
CORRELATIONS VARIABLES=FOOD RENT PUBTRANS TEACHER COOK ENGINEER
  /MISSING=INCLUDE.
```

Overview

`CORRELATIONS` (alias `PEARSON CORR`) produces Pearson product-moment correlations with significance levels and, optionally, univariate statistics, covariances, and cross-product deviations. Other procedures that produce correlation matrices are `PARTIAL CORR`, `REGRESSION`, `DISCRIMINANT`, and `FACTOR`.

Options

Types of Matrices. A simple variable list on the `VARIABLES` subcommand produces a square matrix. You can also request a rectangular matrix of correlations between specific pairs of variables or between variable lists using the keyword `WITH` on `VARIABLES`.

Significance Levels. By default, `CORRELATIONS` displays the number of cases and significance levels for each coefficient. Significance levels are based on a two-tailed test. You can request a one-tailed test, and you can display the significance level for each coefficient as an annotation using the `PRINT` subcommand.

Additional Statistics. You can obtain the mean, standard deviation, and number of nonmissing cases for each variable, and the cross-product deviations and covariance for each pair of variables using the `STATISTICS` subcommand.

Matrix Output. You can write matrix materials to a data file using the `MATRIX` subcommand. The matrix materials include the mean, standard deviation, number of cases used to compute each coefficient, and Pearson correlation coefficient for each variable. The matrix data file can be read by several other procedures.

Basic Specification

- The basic specification is the `VARIABLES` subcommand, which specifies the variables to be analyzed.

- By default, `CORRELATIONS` produces a matrix of correlation coefficients. The number of cases and the significance level are displayed for each coefficient. The significance level is based on a two-tailed test.

Subcommand Order

- The `VARIABLES` subcommand must be first.
- The remaining subcommands can be specified in any order.

Operations

- The correlation of a variable with itself is displayed as 1.0000.
- A correlation that cannot be computed is displayed as a period (.).
- `CORRELATIONS` does not execute if string variables are specified on the variable list.
- This procedure uses the multithreaded options specified by `SET THREADS` and `SET MCACHE`.

Limitations

- A maximum of 40 variable lists.
- A maximum of 500 variables total per command.
- A maximum of 250 syntax elements. Each individual occurrence of a variable name, keyword, or special delimiter counts as 1 toward this total. Variables implied by the `TO` keyword do not count toward this total.

Example

```
CORRELATIONS
  /VARIABLES=sales mpg
  /PRINT=TWOTAIL NOSIG
  /MISSING=PAIRWISE .
```

VARIABLES Subcommand

`VARIABLES` specifies the variable list.

- A simple variable list produces a square matrix of correlations of each variable with every other variable.
- Variable lists joined by the keyword `WITH` produce a rectangular correlation matrix. Variables before `WITH` define the rows of the matrix and variables after `WITH` define the columns.
- The keyword `ALL` can be used on the variable list to refer to all user-defined variables.
- You can specify multiple `VARIABLES` subcommands on a single `CORRELATIONS` command. The slash between the subcommands is required; the keyword `VARIABLES` is not.

Example

```
CORRELATIONS VARIABLES=FOOD RENT PUBTRANS TEACHER COOK ENGINEER
  /VARIABLES=FOOD RENT WITH COOK TEACHER MANAGER ENGINEER
  /MISSING=INCLUDE.
```

- The first `VARIABLES` subcommand requests a square matrix of correlation coefficients among the variables *FOOD*, *RENT*, *PUBTRANS*, *TEACHER*, *COOK*, and *ENGINEER*.
- The second `VARIABLES` subcommand requests a rectangular correlation matrix in which *FOOD* and *RENT* are the row variables and *COOK*, *TEACHER*, *MANAGER*, and *ENGINEER* are the column variables.

PRINT Subcommand

PRINT
Controls whether the significance level is based on a one- or two-tailed test and whether the number of cases and the significance level for each correlation coefficient are displayed.

TWOTAIL
Two-tailed test of significance. This test is appropriate when the direction of the relationship cannot be determined in advance, as is often the case in exploratory data analysis. This is the default.

ONETAIL
One-tailed test of significance. This test is appropriate when the direction of the relationship between a pair of variables can be specified in advance of the analysis.

SIG
Do not flag significant values. `SIG` is the default.

NOSIG
Flag significant values. Values significant at the 0.05 level are flagged with a single asterisk; those that are significant at the 0.01 level are flagged with two asterisks.

FULL
Show the full table. The full correlation matrix table is presented in the output.

LOWER
Show only the lower triangle. Only the correlation matrix table's lower triangle is presented in the output. The keyword allows table output to adhere to APA style guidelines.

STATISTICS Subcommand

The correlation coefficients are automatically displayed in the Correlations table for an analysis specified by a `VARIABLES` list. `STATISTICS` requests additional statistics.

DESCRIPTIVES . *Display mean, standard deviation, and number of nonmissing cases for each variable on the Variables list in the Descriptive Statistics table.* This table precedes all Correlations tables. Variables specified on more than one `VARIABLES` list are displayed only once. Missing values are handled on a variable-by-variable basis regardless of the missing-value option in effect for the correlations.

XPROD . *Display cross-product deviations and covariance for each pair of variables in the Correlations table(s).*

ALL . *All additional statistics.* This produces the same statistics as `DESCRIPTIVES` and `XPROD` together.

MISSING Subcommand

`MISSING` controls the treatment of missing values.

- The `PAIRWISE` and `LISTWISE` keywords are alternatives; however, each can be specified with `INCLUDE` or `EXCLUDE`.
- The default is `PAIRWISE` and `EXCLUDE`.

PAIRWISE . *Exclude missing values pairwise.* Cases that have missing values for one or both of a pair of variables for a specific correlation coefficient are excluded from the computation of that coefficient. Since each coefficient is based on all cases that have valid values for that particular pair of variables, this can result in a set of coefficients based on a varying number of cases. The valid number of cases is displayed in the Correlations table. This is the default.

LISTWISE . *Exclude missing values listwise.* Cases that have missing values for any variable named on any `VARIABLES` list are excluded from the computation of all coefficients across lists. The valid number of cases is the same for all analyses and is displayed in a single annotation.

INCLUDE . *Include user-missing values.* User-missing values are included in the analysis.

EXCLUDE . *Exclude all missing values.* Both user- and system-missing values are excluded from the analysis.

MATRIX Subcommand

`MATRIX` writes matrix materials to a data file or previously declared dataset (`DATASET DECLARE` command). The matrix materials include the mean and standard deviation for each variable, the number of cases used to compute each coefficient, and the Pearson correlation coefficients. Several procedures can read matrix materials produced by `CORRELATIONS`, including `PARTIAL CORR`, `REGRESSION`, `FACTOR`, and `CLUSTER`.

- `CORRELATIONS` cannot write rectangular matrices (those specified with the keyword `WITH`) to a file.

- If you specify more than one variable list on `CORRELATIONS`, only the last list that does not use the keyword `WITH` is written to the matrix data file.
- The keyword `OUT` specifies the file to which the matrix is written. Specify an asterisk to replace the active dataset or a quoted file specification or dataset name, enclosed in parentheses.
- Documents from the original file will not be included in the matrix file and will not be present if the matrix file becomes the working data file.

Format of the Matrix Data File

- The matrix data file has two special variables created by the program: *ROWTYPE_* and *VARNAME_*. The variable *ROWTYPE_* is a short string variable with values *MEAN*, *STDDEV*, *N*, and *CORR* (for Pearson correlation coefficient). The next variable, *VARNAME_*, is a short string variable whose values are the names of the variables used to form the correlation matrix. When *ROWTYPE_* is *CORR*, *VARNAME_* gives the variable associated with that row of the correlation matrix.
- The remaining variables in the file are the variables used to form the correlation matrix.

Split Files

- When split-file processing is in effect, the first variables in the matrix file will be split variables, followed by *ROWTYPE_*, *VARNAME_*, and the variables used to form the correlation matrix.
- A full set of matrix materials is written for each subgroup defined by the split variables.
- A split variable cannot have the same name as any other variable written to the matrix data file.
- If split-file processing is in effect when a matrix is written, the same split-file specifications must be in effect when that matrix is read by another procedure.

Missing Values

- With pairwise treatment of missing values (the default), a matrix of the number of cases used to compute each coefficient is included with the matrix materials.
- With listwise treatment, a single number indicating the number of cases used to calculate all coefficients is included.

Example

```
GET FILE=CITY /KEEP FOOD RENT PUBTRANS TEACHER COOK ENGINEER.
CORRELATIONS VARIABLES=FOOD TO ENGINEER
 /MATRIX OUT(CORRMAT).
```

- `CORRELATIONS` reads data from the file *CITY* and writes one set of matrix materials to the file *CORRMAT*. The working file is still *CITY*. Subsequent commands are executed on *CITY*.

Example

```
GET FILE=CITY /KEEP FOOD RENT PUBTRANS TEACHER COOK ENGINEER.
CORRELATIONS VARIABLES=FOOD TO ENGINEER
 /MATRIX OUT(*).
LIST.
DISPLAY DICTIONARY.
```

- `CORRELATIONS` writes the same matrix as in the example above. However, the matrix data file replaces the working file. The `LIST` and `DISPLAY` commands are executed on the matrix file, not on the *CITY* file.

Example

```
CORRELATIONS VARIABLES=FOOD RENT COOK TEACHER MANAGER ENGINEER
 /FOOD TO TEACHER /PUBTRANS WITH MECHANIC
 /MATRIX OUT(*).
```

- Only the matrix for *FOOD* `TO` *TEACHER* is written to the matrix data file because it is the last variable list that does not use the keyword `WITH`.

MATRIX-END MATRIX

```
MATRIX

matrix statements

END MATRIX
```

The following matrix language statements can be used in a matrix program: `BREAK, CALL, COMPUTE, DISPLAY, DO IF, ELSE, ELSE IF, END IF, END LOOP, GET, LOOP, MGET, MSAVE, PRINT, READ, RELEASE, SAVE, WRITE`.

For information on which functions can be used in matrix language statements, see “Matrix Functions ” on page 57.

Example

```
MATRIX.
READ A /FILE=MATRDATA /SIZE={6,6} /FIELD=1 TO 60.
CALL EIGEN(A,EIGENVEC,EIGENVAL).
LOOP J=1 TO NROW(EIGENVAL).
+ DO IF (EIGENVAL(J) > 1.0).
+   PRINT EIGENVAL(J) / TITLE="Eigenvalue:" /SPACE=3.
+   PRINT T(EIGENVEC(:,J)) / TITLE="Eigenvector:" /SPACE=1.
+ END IF.
END LOOP.
END MATRIX.
```

Release History

Statistics 27.0

- The **`NCDF.BETA`** cumulative distribution function is now supported.
- Probability density functions are now supported (they were previously only supported by the **`COMPUTE`** command).
- Tail probability functions are now supported (they were previously only supported by the **`COMPUTE`** command).
- Random variable functions are now supported (they were previously only supported by the **`COMPUTE`** command).

Subscription November 2017 update / Statistics 26.0

- Long variable names (up to 64 bytes) can be used to name a matrix or vector name (such as `COMPUTE`, `CALL`, `PRINT`, `READ`, `WRITE`, `GET`, `SAVE`, `MGET`, `MSAVE`, `DISPLAY`, `RELEASE`, and so on).
- Variable names that are included in a vector or matrix object are truncated to 8 bytes. This is because the matrix/vector structure is an array of numbers, and each number can match a string only up to 8 bytes. Long names (up to 64 bytes) are supported only when explicitly specified.
- Long variable names are supported in `GET` and `SAVE` commands when explicitly specified on the `/VARIABLES` subcommand (and when specified on the `/STRINGS` subcommand for the `SAVE` command). Variable names for `GET` and `SAVE` commands are truncated to 8 bytes when they are referenced through a vector in the `/NAMES` subcommand.
- The `GET`, `SAVE`, `MGET`, or `MSAVE` statements support both dataset references and physical file specifications.
- **`MATRIX-END MATRIX`** supports statistical functions that were previously only supported by the **`COMPUTE`** command (for example **`IDF.CHISQ`**, **`CDF.NORMAL`**, **`NCDF.F`**, and so on). For more information, see Matrix Functions (MATRIX-END MATRIX command).

Overview

The `MATRIX` and `END MATRIX` commands enclose statements that are executed by the matrix processor. Using matrix programs, you can write your own statistical routines in the compact language of matrix algebra. Matrix programs can include mathematical calculations, control structures, display of results, and reading and writing matrices as character files or data files.

As discussed below, a matrix program is for the most part independent of the rest of the session, although it can read and write IBM SPSS Statistics data files, including the active dataset.

This section does not attempt to explain the rules of matrix algebra. Many textbooks teach the application of matrix methods to statistics.

The `MATRIX` procedure was originally developed at the Madison Academic Computing Center, University of Wisconsin.

Using open datasets

Use of datasets that are currently open in a session is not supported, except as specifically noted for the active dataset.

Terminology

A variable within a matrix program represents a **matrix**, which is simply a set of values arranged in a rectangular array of rows and columns.

- An $n \times m$ (read "n by m") matrix is one that has n rows and m columns. The integers n and m are the dimensions of the matrix. An $n \times m$ matrix contains $n \times m$ elements, or data values.
- An $n \times 1$ matrix is sometimes called a **column vector**, and a $1 \times n$ matrix is sometimes called a **row vector**. A vector is a special case of a matrix.
- A 1 × 1 matrix, containing a single data value, is often called a **scalar**. A scalar is also a special case of a matrix.
- An **index** to a matrix or vector is an integer that identifies a specific row or column. Indexes normally appear in printed works as subscripts, as in A_{31}, but are specified in the matrix language within parentheses, as in $A(3,1)$. The row index for a matrix precedes the column index.
- The **main diagonal** of a matrix consists of the elements whose row index equals their column index. It begins at the top left corner of the matrix; in a square matrix, it runs to the bottom right corner.
- The **transpose** of a matrix is the matrix with rows and columns interchanged. The transpose of an $n \times m$ matrix is an $m \times n$ matrix.
- A **symmetric matrix** is a square matrix that is unchanged if you flip it about the main diagonal. That is, the element in row i, column j equals the element in row j, column i. A symmetric matrix equals its transpose.
- Matrices are always rectangular, although it is possible to read or write symmetric matrices in triangular form. Vectors and scalars are considered degenerate rectangles.
- It is an error to try to create a matrix whose rows have different numbers of elements.

A matrix program does not process individual cases unless you so specify, using the control structures of the matrix language. Unlike ordinary IBM SPSS Statistics variables, matrix variables do not have distinct values for different cases. A matrix is a single entity.

Vectors in matrix processing should not be confused with the vectors temporarily created by the `VECTOR` command. The latter are shorthand for a list of IBM SPSS Statistics variables and, like all ordinary IBM SPSS Statistics variables, are unavailable during matrix processing.

Matrix Variables

A matrix variable is created by a matrix statement that assigns a value to a variable name.

- A matrix variable name follows the same rules as those applicable to an ordinary IBM SPSS Statistics variable name.
- The names of matrix functions and procedures cannot be used as variable names within a matrix program. (In particular, the letter *T* cannot be used as a variable name because T is an alias for the `TRANSPOS` function.)
- The `COMPUTE`, `READ`, `GET`, `MGET`, and `CALL` statements create matrices. An index variable named on a `LOOP` statement creates a scalar with a value assigned to it.

- A variable name can be redefined within a matrix program without regard to the dimensions of the matrix it represents. The same name can represent scalars, vectors, and full matrices at different points in the matrix program.
- `MATRIX-END MATRIX` does not include any special processing for missing data. When reading a data matrix from a data file, you must therefore specify whether missing data are to be accepted as valid or excluded from the matrix.

String Variables in Matrix Programs

Matrix variables can contain short string data. Support for string variables is limited, however.

- `MATRIX` will attempt to carry out calculations with string variables if you so request. The results will not be meaningful.
- You must specify a format (such as A8) when you display a matrix that contains string data.

Support for Long Variable Names

- Long variable names (up to 64 bytes) can be used to name a matrix, vector, or scalar (such as `COMPUTE`, `CALL`, `PRINT`, `READ`, `WRITE`, `GET`, `SAVE`, `MGET`, `MSAVE`, `DISPLAY`, `RELEASE`, and so on).
- Variable names that are included in a vector or matrix object are truncated to 8 bytes. This is because the matrix/vector structure is an array of numbers, and each number can match a string only up to 8 bytes. Long names (up to 64 bytes) are supported only when explicitly specified.
- Long variable names are supported in `GET` and `SAVE` commands when explicitly specified on the `/VARIABLES` subcommand (and when specified on the `/STRINGS` subcommand for the SAVE command). Variable names for **GET** and **SAVE** commands are truncated to 8 bytes when they are referenced through a vector in the `/NAMES` subcommand.

Syntax of Matrix Language

A matrix program consists of statements. Matrix statements must appear in a matrix program, between the `MATRIX` and `END MATRIX` commands. They are analogous to IBM SPSS Statistics commands and follow the rules of the command language regarding the abbreviation of keywords; the equivalence of upper and lower case; the use of spaces, commas, and equals signs; and the splitting of statements across multiple lines. However, commas are required to separate arguments to matrix functions and procedures and to separate variable names on the `RELEASE` statement.

Matrix statements are composed of the following elements:

- Keywords, such as the names of matrix statements
- Variable names
- Explicitly written matrices, which are enclosed within braces ({})
- Arithmetic and logical operators
- Matrix functions
- The command terminator, which serves as a statement terminator within a matrix program

Comments in Matrix Programs

Within a matrix program, you can enter comments in any of the following forms: on lines beginning with the `COMMENT` command, on lines beginning with an asterisk, or between the characters /* and */ on a command line.

Matrix Notation

To write a matrix explicitly:

- Enclose the matrix within braces ({}).
- Separate the elements of each row by commas.
- Separate the rows by semicolons.

- String elements must be enclosed in either quotes, as is generally true in the command language.

Matrix Notation Shorthand

You can simplify the construction of matrices using notation shorthand.

Consecutive Integers. Use a colon to indicate a range of consecutive integers. For example, the vector `{1,2,3,4,5,6}` can be written as `{1:6}`.

Incremented Ranges of Integers. Use a second colon followed by an integer to indicate the increment. The matrix `{1,3,5,7;2,5,8,11}` can be written as `{1:7:2;2:11:3}`, where `1:7:2` indicates the integers from 1 to 7 incrementing by 2, and `2:11:3` indicates the integers from 2 to 11 incrementing by 3.

- You must use integers when specifying a range in either of these ways. Numbers with fractional parts are truncated to integers.
- If an arithmetic expression is used, it should be enclosed in parentheses.

Construction of a Matrix from Other Matrices

You can use vector or rectangular matrices to construct a new matrix, separating row expressions by semicolons and components of row expressions by commas. If a column vector V_c has n elements and matrix M has the dimensions $n \times m$, then $\{M; V_c\}$ is an $n \times (m + 1)$ matrix. Similarly, if the row vector V_r has m elements and M is the same, then $\{M; V_r\}$ is an $(n + 1) \times m$ matrix. In fact, you can paste together any number of matrices and vectors this way.

- All of the components of each column expression must have the same number of actual rows, and all of the row expressions must have the same number of actual columns.
- The distinction between row vectors and column vectors must be observed carefully when constructing matrices in this way, so that the components will fit together properly.
- Several of the matrix functions are also useful in constructing matrices; see in particular the `MAKE`, `UNIFORM`, and `IDENT` functions in "Matrix Functions " on page 57 .

Matrix Operations

You can perform matrix calculations according to the rules of matrix algebra and compare matrices using relational or logical operators.

Conformable Matrices

Many operations with matrices make sense only if the matrices involved have "suitable" dimensions. Most often, this means that they should be the same size, with the same number of rows and the same number of columns. Matrices that are the right size for an operation are said to be **conformable matrices**. If you attempt to do something in a matrix program with a matrix that is not conformable for that operation--a matrix that has the wrong dimensions--you will receive an error message, and the operation will not be performed. An important exception, where one of the matrices is a scalar, is discussed below.

Requirements for carrying out matrix operations include:

- Matrix addition and subtraction require that the two matrices be the same size.
- The relational and logical operations described below require that the two matrices be the same size.
- Matrix multiplication requires that the number of columns of the first matrix equal the number of rows of the second matrix.
- Raising a matrix to a power can be done only if the matrix is square. This includes the important operation of *inverting* a matrix, where the power is −1.
- Conformability requirements for matrix functions are noted in "Matrix Functions " on page 57 and in "COMPUTE Statement " on page 56 .

Scalar Expansion

When one of the matrices involved in an operation is a scalar, the scalar is treated as a matrix of the correct size in order to carry out the operation. This internal scalar expansion is performed for the following operations:

- Addition and subtraction.
- Elementwise multiplication, division, and exponentiation. Note that multiplying a matrix elementwise by an expanded scalar is equivalent to ordinary scalar multiplication--each element of the matrix is multiplied by the scalar.
- All relational and logical operators.

Arithmetic Operators

You can add, subtract, multiply, or exponentiate matrices according to the rules of matrix algebra, or you can perform elementwise arithmetic, in which you multiply, divide, or exponentiate each element of a matrix separately. The arithmetic operators are listed below.

`Unary –`. *Sign reversal.* A minus sign placed in front of a matrix reverses the sign of each element. (The unary + is also accepted but has no effect.)

`+`. *Matrix addition.* Corresponding elements of the two matrices are added. The matrices must have the same dimensions, or one must be a scalar.

`–`. *Matrix subtraction.* Corresponding elements of the two matrices are subtracted. The matrices must have the same dimensions, or one must be a scalar.

`*`. *Multiplication.* There are two cases. First, *scalar multiplication*: if either of the matrices is a scalar, each element of the other matrix is multiplied by that scalar. Second, *matrix multiplication*.

/. *Division.* The division operator performs elementwise division (described below). True matrix division, the inverse operation of matrix multiplication, is accomplished by taking the `INV` function (square matrices) or the `GINV` function (rectangular matrices) of the denominator and multiplying.

`**`. *Matrix exponentiation.* A matrix can be raised only to an integer power. The matrix, which must be square, is multiplied by itself as many times as the absolute value of the exponent. If the exponent is negative, the result is then inverted.

`&*`. *Elementwise multiplication.* Each element of the matrix is multiplied by the corresponding element of the second matrix. The matrices must have the same dimensions, or one must be a scalar.

`&/`. *Elementwise division.* Each element of the matrix is divided by the corresponding element of the second matrix. The matrices must have the same dimensions, or one must be a scalar.

`&**`. *Elementwise exponentiation.* Each element of the first matrix is raised to the power of the corresponding element of the second matrix. The matrices must have the same dimensions, or one must be a scalar.

`:`. *Sequential integers.* This operator creates a vector of consecutive integers from the value preceding the operator to the value following it. You can specify an optional increment following a second colon. See "Matrix Notation Shorthand " on page 52 for the principal use of this operator.

- Use these operators only with numeric matrices. The results are undefined when they are used with string matrices.

Relational Operators

The relational operators are used to compare two matrices, element by element. The result is a matrix of the same size as the (expanded) operands and containing either 1 or 0. The value of each element, 1 or 0, is determined by whether the comparison between the corresponding element of the first matrix with the corresponding element of the second matrix is true or false--1 for true and 0 for false. The matrices being compared must be of the same dimensions unless one of them is a scalar. The relational operators are listed in the following table.

Table 1. Relational operators in matrix programs

Symbol	Keyword	Definition
>	GT	Greater than
<	LT	Less than
<> or ~= (¬=)	NE	Not equal to
<=	LE	Less than or equal to
>=	GE	Greater than or equal to
=	EQ	Equal to

- The symbolic and alphabetic forms of these operators are equivalent.
- The symbols representing NE (~= or ¬=) are system dependent. In general, the tilde (~) is valid for ASCII systems, while the logical-not sign (¬), or whatever symbol is over the number 6 on the keyboard, is valid for IBM EBCDIC systems.
- Use these operators only with numeric matrices. The results are undefined when they are used with string matrices.

Logical Operators

Logical operators combine two matrices, normally containing values of 1 (true) or 0 (false). When used with other numerical matrices, they treat all positive values as true and all negative and 0 values as false. The logical operators are:

NOT. *Reverses the truth of the matrix that follows it.* Positive elements yield 0, and negative or 0 elements yield 1.

AND. *Both must be true.* The matrix *A* `AND` *B* is 1 where the corresponding elements of *A* and *B* are both positive and 0 elsewhere.

OR. *Either must be true.* The matrix *A* `OR` *B* is 1 where the corresponding element of either *A* or *B* is positive and 0 where both elements are negative or 0.

XOR. *Either must be true but not both.* The matrix *A* `XOR` *B* is 1 where one but not both of the corresponding elements of *A* and *B* is positive and 0 where both are positive or neither is positive.

Precedence of Operators

Parentheses can be used to control the order in which complex expressions are evaluated. When the order of evaluation is not specified by parentheses, operations are carried out in the order listed below. The operations higher on the list take precedence over the operations lower on the list.

+ – (unary)

:

** &**

* &* &/

+ – (addition and subtraction)

> >= < <= <>=

NOT

AND

OR XOR

Operations of equal precedence are performed left to right of the expressions.

Examples

```
COMPUTE A = {1,2,3;4,5,6}.
COMPUTE B = A + 4.
COMPUTE C = A &** 2.
COMPUTE D = 2 &** A.
COMPUTE E = A < 5.
COMPUTE F = (C &/ 2) < B.
```

MATRIX and Other Commands

A matrix program is a single procedure within a session.

- No active dataset is needed to run a matrix program. If one exists, it is ignored during matrix processing unless you specifically reference it (with an asterisk) on the `GET`, `SAVE`, `MGET`, or `MSAVE` statements.
- The `GET`, `SAVE`, `MGET`, or `MSAVE` statements support both dataset references and physical file specifications. Take the following example:

```
DATASET CLOSE ALL.
NEW FILE.

GET FILE 'Samples\English\Employee data.sav'.

DATASET DECLARE corr.
CORRELATIONS bdate to minority /MATRIX=out(corr).
MATRIX.
MGET  /file=corr/type=corr.
PRINT CR.
END MATRIX.
```

The example uses `corr` as a reference to a virtual dataset. The correlation output can also be saved to a physical file.

- Variables defined in the active dataset are unavailable during matrix processing, except with the `GET` or `MGET` statements.
- Matrix variables are unavailable after the `END MATRIX` command unless you use `SAVE` or `MSAVE` to write them to the active dataset.
- You cannot run a matrix program from a syntax window if split-file processing is in effect. If you save the matrix program into a syntax file, however, you can use the `INCLUDE` command to run the program even if split-file processing is in effect.

Matrix Statements

Most valid matrix statements have the same name as an analogous IBM SPSS Statistics command and perform an exactly analogous function. Use only these statements between the `MATRIX` and `END MATRIX` commands. Any command not recognized as a valid matrix statement will be rejected by the matrix processor.

Valid matrix statements are: `BREAK, CALL, COMPUTE, DISPLAY, DO IF, ELSE, ELSE IF, END IF, END LOOP, GET, LOOP, MGET, MSAVE, PRINT, READ, RELEASE, SAVE, WRITE`.

There is a maximum of 100 SAVE commands in a matrix program.

Exchanging Data with IBM SPSS Statistics Data Files

Matrix programs can read and write IBM SPSS Statistics data files.

- The `GET` and `SAVE` statements read and write ordinary (case-oriented) IBM SPSS Statistics data files, treating each case as a row of a matrix and each ordinary variable as a column.

A matrix program cannot contain more than 100 SAVE commands.

- The `MGET` and `MSAVE` statements read and write matrix-format data files, respecting the structure defined by IBM SPSS Statistics when it creates the file.
- Case weighting in a data file is ignored when the file is read into a matrix program.

MATRIX and END MATRIX Commands

The `MATRIX` command, when encountered in a session, invokes the matrix processor, which reads matrix statements until the `END MATRIX` or `FINISH` command is encountered.

- `MATRIX` is a procedure and cannot be entered inside a transformation structure such as `DO IF` or `LOOP`.
- The `MATRIX` procedure does not require an active dataset.
- Comments are removed before subsequent lines are passed to the matrix processor.
- Macros are expanded before subsequent lines are passed to the matrix processor.

The `END MATRIX` command terminates matrix processing and returns control to the command processor.

- The contents of matrix variables are lost after an `END MATRIX` command.
- The active dataset, if present, becomes available again after an `END MATRIX` command.

COMPUTE Statement

The `COMPUTE` statement carries out most of the calculations in the matrix program. It closely resembles the `COMPUTE` command in the IBM SPSS Statistics transformation language.

- The basic specification is the target variable, an equals sign, and the assignment expression. Values of the target variable are calculated according to the specification on the assignment expression.
- The target variable must be named first, and the equals sign is required. Only one target variable is allowed per `COMPUTE` statement.
- Expressions that extract portions of a matrix, such as *M*(1,:) or *M*(1:3,4), are allowed to assign values. (See the topic "Matrix Notation Shorthand " on page 52 for more information.) The target variable must be specified as a variable.
- Matrix functions must specify at least one argument enclosed in parentheses. If an expression has two or more arguments, each argument must be separated by a comma. For a complete discussion of the functions and their arguments, see "Matrix Functions " on page 57 .

String Values on COMPUTE Statements

Matrix variables, unlike those in the transformation language, are not checked for data type (numeric or string) when you use them in a `COMPUTE` statement.

- Numerical calculations with matrices containing string values will produce meaningless results.
- One or more elements of a matrix can be set equal to string constants by enclosing the string constants in quotes on a `COMPUTE` statement.
- String values can be copied from one matrix to another with the `COMPUTE` statement.
- There is no way to display a matrix that contains both numeric and string values, if you compute one for some reason.

Example

```
COMPUTE LABELS={"Observe", "Predict", "Error"}.
PRINT LABELS /FORMAT=A7.
```

- *LABELS* is a row vector containing three string values.

Arithmetic Operations and Comparisons

The expression on a `COMPUTE` statement can be formed from matrix constants and variables, combined with the arithmetic, relational, and logical operators. Matrix constructions and matrix functions are also allowed.

Examples

```
COMPUTE PI = 3.14159265.
COMPUTE RSQ = R * R.
COMPUTE FLAGS = EIGENVAL >= 1.
COMPUTE ESTIM = {OBS, PRED, ERR}.
```

- The first statement computes a scalar. Note that the braces are optional on a scalar constant.
- The second statement computes the square of the matrix *R*. *R* can be any square matrix, including a scalar.
- The third statement computes a vector named *FLAGS*, which has the same dimension as the existing vector *EIGENVAL*. Each element of *FLAGS* equals 1 if the corresponding element of *EIGENVAL* is greater than or equal to 1, and 0 if the corresponding element is less than 1.
- The fourth statement constructs a matrix *ESTIM* by concatenating the three vectors or matrices *OBS*, *PRED*, and *ERR*. The component matrices must have the same number of rows.

Matrix Functions

The following functions are available in the matrix program. Except where noted, each takes one or more numeric matrices as arguments and returns a matrix value as its result. The arguments must be enclosed in parentheses, and multiple arguments must be separated by commas.

On the following list, matrix arguments are represented by names beginning with *M*. Unless otherwise noted, these arguments can be vectors or scalars. Arguments that must be vectors are represented by names beginning with *V*, and arguments that must be scalars are represented by names beginning with *S*.

ABS(M). *Absolute value.* Takes a single argument. Returns a matrix having the same dimensions as the argument, containing the absolute values of its elements.

ALL(M). *Test for all elements nonzero.* Takes a single argument. Returns a scalar: 1 if all elements of the argument are nonzero and 0 if any element is zero.

ANY(M). *Test for any element nonzero.* Takes a single argument. Returns a scalar: 1 if any element of the argument is nonzero and 0 if all elements are zero.

ARSIN(M). *Inverse sine.* Takes a single argument, whose elements must be between −1 and 1. Returns a matrix having the same dimensions as the argument, containing the inverse sines (arcsines) of its elements. The results are in radians and are in the range from −π/2 to π/2.

ARTAN(M). *Inverse tangent.* Takes a single argument. Returns a matrix having the same dimensions as the argument, containing the inverse tangents (arctangents) of its elements, in radians. To convert radians to degrees, multiply by 180/π, which you can compute as 45/ARTAN(1). For example, the statement `COMPUTE DEGREES=ARTAN(M)*45/ARTAN(1)` returns a matrix containing inverse tangents in degrees.

BLOCK(M1,M2,...). *Create a block diagonal matrix.* Takes any number of arguments. Returns a matrix with as many rows as the sum of the rows in all the arguments, and as many columns as the sum of the columns in all the arguments, with the argument matrices down the diagonal and zeros elsewhere.

CHOL(M). *Cholesky decomposition.* Takes a single argument, which must be a symmetric positive-definite matrix (a square matrix, symmetric about the main diagonal, with positive eigenvalues). Returns a matrix having the same dimensions as the argument. If *M* is a symmetric positive-definite matrix and *B*=`CHOL`(*M*), then T(*B*)* *B*=*M*, where T is the transpose function defined below.

CMAX(M). *Column maxima.* Takes a single argument. Returns a row vector with the same number of columns as the argument. Each column of the result contains the maximum value of the corresponding column of the argument.

CMIN(M). *Column minima.* Takes a single argument. Returns a row vector with the same number of columns as the argument. Each column of the result contains the minimum value of the corresponding column of the argument.

COS(M). *Cosines.* Takes a single argument. Returns a matrix having the same dimensions as the argument, containing the cosines of the elements of the argument. Elements of the argument matrix are assumed to be measured in radians. To convert degrees to radians, multiply by π/180, which you can compute as ARTAN(1)/45. For example, the statement `COMPUTE COSINES=COS(DEGREES*ARTAN(1)/45)` returns cosines from a matrix containing elements measured in degrees.

CSSQ(M). *Column sums of squares.* Takes a single argument. Returns a row vector with the same number of columns as the argument. Each column of the result contains the sum of the squared values of the elements in the corresponding column of the argument.

CSUM(M). *Column sums.* Takes a single argument. Returns a row vector with the same number of columns as the argument. Each column of the result contains the sum of the elements in the corresponding column of the argument.

DESIGN(M). *Main-effects design matrix from the columns of a matrix.* Takes a single argument. Returns a matrix having the same number of rows as the argument, and as many columns as the sum of the numbers of unique values in each column of the argument. Constant columns in the argument are skipped with a warning message. The result contains 1 in the row(s) where the value in question occurs in the argument and 0 otherwise.

DET(M). *Determinant.* Takes a single argument, which must be a square matrix. Returns a scalar, which is the determinant of the argument.

DIAG(M). *Diagonal of a matrix.* Takes a single argument. Returns a column vector with as many rows as the minimum of the number of rows and the number of columns in the argument. The *i*th element of the result is the value in row *i*, column *i* of the argument.

EOF(file). *End of file indicator.* Normally used after a `READ` statement. Takes a single argument, which must be either a filename in quotes, or a file handle defined on a `FILE HANDLE` command that precedes the matrix program. Returns a scalar equal to 1 if the last attempt to read that file encountered the last record in the file, and equal to 0 if the last attempt did not encounter the last record in the file. Calling the `EOF` function causes a `REREAD` specification on the `READ` statement to be ignored on the next attempt to read the file.

EVAL(M). *Eigenvalues of a symmetric matrix.* Takes a single argument, which must be a symmetric matrix. Returns a column vector with the same number of rows as the argument, containing the eigenvalues of the argument in decreasing numerical order.

EXP(M). *Exponentials of matrix elements.* Takes a single argument. Returns a matrix having the same dimensions as the argument, in which each element equals *e* raised to the power of the corresponding element in the argument matrix.

GINV(M). *Moore-Penrose generalized inverse of a matrix.* Takes a single argument. Returns a matrix with the same dimensions as the transpose of the argument. If *A* is the generalized inverse of a matrix *M*, then *M*A*M=M* and *A*M*A=A*. Both *A*M* and *M*A* are symmetric.

GRADE(M). *Ranks elements in a matrix*. Takes a single argument. Uses sequential integers for ties.

GSCH(M). *Gram-Schmidt orthonormal basis for the space spanned by the column vectors of a matrix.* Takes a single argument, in which there must be as many linearly independent columns as there are rows. (That is, the rank of the argument must equal the number of rows.) Returns a square matrix with as many rows as the argument. The columns of the result form a basis for the space spanned by the columns of the argument.

IDENT(S1 [,S2]). *Create an identity matrix.* Takes either one or two arguments, which must be scalars. Returns a matrix with as many rows as the first argument and as many columns as the second argument, if any. If the second argument is omitted, the result is a square matrix. Elements on the main diagonal of the result equal 1, and all other elements equal 0.

INV(M). *Inverse of a matrix.* Takes a single argument, which must be square and nonsingular (that is, its determinant must not be 0). Returns a square matrix having the same dimensions as the argument. If *A* is the inverse of *M*, then *M*A=A*M=I*, where *I* is the identity matrix.

KRONEKER(M1,M2).

A(1,1)**B* *A*(1,2)* *B* ... *A*(1,*N*)**B*

A(2,1)**B* *A*(2,2)* *B* ... *A*(2,*N*)* *B*

...

A(*M*,1)**B* *A*(*M*,2)**B* ... *A*(*M*, *N*)**B*

LG10(M). *Base 10 logarithms of the elements.* Takes a single argument, all of whose elements must be positive. Returns a matrix having the same dimensions as the argument, in which each element is the logarithm to base 10 of the corresponding element of the argument.

LN(M). *Natural logarithms of the elements.* Takes a single argument, all of whose elements must be positive. Returns a matrix having the same dimensions as the argument, in which each element is the logarithm to base *e* of the corresponding element of the argument.

MAGIC(S). *Magic square.* Takes a single scalar, which must be 3 or larger, as an argument. Returns a square matrix with *S* rows and *S* columns containing the integers from 1 through S^2. All of the row sums and all of the column sums are equal in the result matrix. (The result matrix is only one of several possible magic squares.)

MAKE(S1,S2,S3). *Create a matrix, all of whose elements equal a specified value.* Takes three scalars as arguments. Returns an $S1 \times S2$ matrix, all of whose elements equal $S3$.

MDIAG(V). *Create a square matrix with a specified main diagonal.* Takes a single vector as an argument. Returns a square matrix with as many rows and columns as the dimension of the vector. The elements of the vector appear on the main diagonal of the matrix, and the other matrix elements are all 0.

MMAX(M). *Maximum element in a matrix.* Takes a single argument. Returns a scalar equal to the numerically largest element in the argument *M*.

MMIN(M). *Minimum element in a matrix.* Takes a single argument. Returns a scalar equal to the numerically smallest element in the argument *M*.

MOD(M,S). *Remainders after division by a scalar.* Takes two arguments, a matrix and a scalar (which must not be 0). Returns a matrix having the same dimensions as *M*, each of whose elements is the remainder after the corresponding element of *M* is divided by *S*. The sign of each element of the result is the same as the sign of the corresponding element of the matrix argument *M*.

MSSQ(M). *Matrix sum of squares.* Takes a single argument. Returns a scalar that equals the sum of the squared values of all of the elements in the argument.

MSUM(M). *Matrix sum.* Takes a single argument. Returns a scalar that equals the sum of all of the elements in the argument.

NCOL(M). *Number of columns in a matrix.* Takes a single argument. Returns a scalar that equals the number of columns in the argument.

NROW(M). *Number of rows in a matrix.* Takes a single argument. Returns a scalar that equals the number of rows in the argument.

RANK(M). *Rank of a matrix.* Takes a single argument. Returns a scalar that equals the number of linearly independent rows or columns in the argument.

RESHAPE(M,S1,S2). *Matrix of different dimensions.* Takes three arguments, a matrix and two scalars, whose product must equal the number of elements in the matrix. Returns a matrix whose dimensions are given by the scalar arguments. For example, if *M* is any matrix with exactly 50 elements, then `RESHAPE(M, 5, 10)` is a matrix with 5 rows and 10 columns. Elements are assigned to the reshaped matrix in order by row.

RMAX(M). *Row maxima.* Takes a single argument. Returns a column vector with the same number of rows as the argument. Each row of the result contains the maximum value of the corresponding row of the argument.

RMIN(M). *Row minima.* Takes a single argument. Returns a column vector with the same number of rows as the argument. Each row of the result contains the minimum value of the corresponding row of the argument.

RND(M). *Elements rounded to the nearest integers.* Takes a single argument. Returns a matrix having the same dimensions as the argument. Each element of the result equals the corresponding element of the argument rounded to an integer.

RNKORDER(M). *Ranking of matrix elements in ascending order.* Takes a single argument. Returns a matrix having the same dimensions as the argument *M*. The smallest element of the argument corresponds to a result element of 1, and the largest element of the argument to a result element equal to the number of elements, except that ties (equal elements in *M*) are resolved by assigning a rank equal to the arithmetic mean of the applicable ranks.

RSSQ(M). *Row sums of squares.* Takes a single argument. Returns a column vector having the same number of rows as the argument. Each row of the result contains the sum of the squared values of the elements in the corresponding row of the argument.

RSUM(M). *Row sums.* Takes a single argument. Returns a column vector having the same number of rows as the argument. Each row of the result contains the sum of the elements in the corresponding row of the argument.

SIN(M). *Sines.* Takes a single argument. Returns a matrix having the same dimensions as the argument, containing the sines of the elements of the argument. Elements of the argument matrix are assumed to be measured in radians. To convert degrees to radians, multiply by π/180, which you can compute as ARTAN(1)/45. For example, the statement `COMPUTE SINES=SIN(DEGREES*ARTAN(1)/45)` computes sines from a matrix containing elements measured in degrees.

SOLVE(M1,M2). *Solution of systems of linear equations.* Takes two arguments, the first of which must be square and nonsingular (its determinant must be nonzero), and the second of which must have the same number of rows as the first. Returns a matrix with the same dimensions as the second argument. If *M1*X=M2*, then *X*= `SOLVE`(*M1, M2*). In effect, this function sets its result *X* equal to `INV`(*M1*)**M2*.

SQRT(M). *Square roots of elements.* Takes a single argument whose elements must not be negative. Returns a matrix having the same dimensions as the arguments, whose elements are the positive square roots of the corresponding elements of the argument.

SSCP(M). *Sums of squares and cross-products.* Takes a single argument. Returns a square matrix having as many rows (and columns) as the argument has columns. SSCP *(M)* equals T *(M)*M*, where T is the transpose function defined below.

SVAL(M). *Singular values of a matrix.* Takes a single argument. Returns a column vector containing as many rows as the minimum of the numbers of rows and columns in the argument, containing the singular values of the argument in decreasing numerical order. The singular values of a matrix *M* are the square roots of the eigenvalues of T *(M)*M*, where T is the transpose function discussed below.

SWEEP(M,S).

Sweep transformation of a matrix. Takes two arguments, a matrix and a scalar, which must be less than or equal to both the number of rows and the number of columns of the matrix. In other words, the pivot element of the matrix, which is *M*(*S*,*S*), must exist. Returns a matrix of the same dimensions as *M*. Suppose that *S*={ *k*} and *A* = `SWEEP`(*M*,*S*). If *M*(*k,k*) is not 0, then

A(*k*,*k*) = 1/*M*(*k*,*k*)

A(*i*,*k*) = −*M*(*i*,*k*)/*M*(*k*,*k*), for *i* not equal to *k*

A(*k*,*j*) = *M*(*k*,*j*)/*M*(*k*,*k*), for *j* not equal to *k*

A(*i*,*j*) = (*M*(*i*,*j*)**M*(*k*,*k*), − *M*(*i*,*k*)**M*(*k*,*j*))/*M*(*k*,*k*), for *i,j* not equal to *k*

and if *M*(*k,k*) equals 0, then

A(*i*,*k*) = *A*(*k*,*i*) = 0, for all *i*

A(*i*,*j*) = *M*(*i*,*j*), for *i,j* not equal to *k*

TRACE(M). *Sum of the main diagonal elements.* Takes a single argument. Returns a scalar, which equals the sum of the elements on the main diagonal of the argument.

TRANSPOS(M). *Transpose of the matrix.* Takes a single argument. Returns the transpose of the argument. `TRANSPOS` can be shortened to T.

TRUNC(M). *Truncation of elements to integers.* Takes a single argument. Returns a matrix having the same dimensions as the argument, whose elements equal the corresponding elements of the argument truncated to integers.

UNIFORM(S1,S2). *Uniformly distributed pseudo-random numbers between 0 and 1.* Takes two scalars as arguments. Returns a matrix with the number of rows specified by the first argument and the number of columns specified by the second argument, containing pseudo-random numbers uniformly distributed between 0 and 1. The function respects the traditional versus twister and seed settings for the corresponding generator.

MATRIX probability density functions

The following functions give the value of the density function with the specified distribution at the value *quant*, the first argument. All functions accept matrix, vector, or scalar as the first argument. The **`PDF.BVNOR`** function accepts only scalar as the third argument. The result has the same dimension as the first argument. Note the period in each function name.

PDF.BERNOULLI. PDF.BERNOULLI(quant, prob). Numeric. Returns the probability that a value from the Bernoulli distribution, with the given probability parameter, will be equal to quant.

PDF.BETA. PDF.BETA(quant, shape1, shape2). Numeric. Returns the probability density of the beta distribution, with the given shape parameters, at quant.

PDF.BINOM. PDF.BINOM(quant, n, prob). Numeric. Returns the probability that the number of successes in n trials, with probability prob of success in each, will be equal to quant. When n is 1, this is the same as PDF.BERNOULLI.

PDF.BVNOR. PDF.BVNOR(quant1, quant2, corr). Numeric. Returns the probability density of the standard bivariate normal distribution, with the given correlation parameter, at quant1, quant2.

PDF.CAUCHY. PDF.CAUCHY(quant, loc, scale). Numeric. Returns the probability density of the Cauchy distribution, with the given location and scale parameters, at quant.

PDF.CHISQ. PDF.CHISQ(quant, df). Numeric. Returns the probability density of the chi-square distribution, with df degrees of freedom, at quant.

PDF.EXP. PDF.EXP(quant, shape). Numeric. Returns the probability density of the exponential distribution, with the given shape parameter, at quant.

PDF.F. PDF.F(quant, df1, df2). Numeric. Returns the probability density of the F distribution, with degrees of freedom df1 and df2, at quant.

PDF.GAMMA. PDF.GAMMA(quant, shape, scale). Numeric. Returns the probability density of the gamma distribution, with the given shape and scale parameters, at quant.

PDF.GEOM. PDF.GEOM(quant, prob). Numeric. Returns the probability that the number of trials to obtain a success, when the probability of success is given by prob, will be equal to quant.

PDF.HALFNRM. PDF.HALFNRM(quant, mean, stddev). Numeric. Returns the probability density of the half normal distribution, with specified mean and standard deviation, at quant.

PDF.HYPER. PDF.HYPER(quant, total, sample, hits). Numeric. Returns the probability that the number of objects with a specified characteristic, when sample objects are randomly selected from a universe of size total in which hits have the specified characteristic, will be equal to quant.

PDF.IGAUSS. PDF.IGAUSS(quant, loc, scale). Numeric. Returns the probability density of the inverse Gaussian distribution, with the given location and scale parameters, at quant.

PDF.LAPLACE. PDF.LAPLACE(quant, mean, scale). Numeric. Returns the probability density of the Laplace distribution, with the specified mean and scale parameters, at quant.

PDF.LOGISTIC. PDF.LOGISTIC(quant, mean, scale). Numeric. Returns the probability density of the logistic distribution, with the specified mean and scale parameters, at quant.

PDF.LNORMAL. PDF.LNORMAL(quant, a, b). Numeric. Returns the probability density of the log-normal distribution, with the specified parameters, at quant.

PDF.NEGBIN. PDF.NEGBIN(quant, thresh, prob). Numeric. Returns the probability that the number of trials to obtain a success, when the threshold parameter is thresh and the probability of success is given by prob, will be equal to quant.

PDF.NORMAL. PDF.NORMAL(quant, mean, stddev). Numeric. Returns the probability density of the normal distribution, with specified mean and standard deviation, at quant.

PDF.PARETO. PDF.PARETO(quant, threshold, shape). Numeric. Returns the probability density of the Pareto distribution, with the specified threshold and shape parameters, at quant.

PDF.POISSON. PDF.POISSON(quant, mean). Numeric. Returns the probability that a value from the Poisson distribution, with the specified mean or rate parameter, will be equal to quant.

PDF.T. PDF.T(quant, df). Numeric. Returns the probability density of Student's t distribution, with the specified degrees of freedom df, at quant.

PDF.UNIFORM. PDF.UNIFORM(quant, min, max). Numeric. Returns the probability density of the uniform distribution, with the specified minimum and maximum, at quant.

PDF.WEIBULL. PDF.WEIBULL(quant, a, b). Numeric. Returns the probability density of the Weibull distribution, with the specified parameters, at quant.

NPDF.BETA. NPDF.BETA(quant, shape1, shape2, nc). Numeric. Returns the probability density of the noncentral beta distribution, with the given shape and noncentrality parameters, at quant.

NPDF.CHISQ. NPDF.CHISQ(quant, df, nc). Numeric. Returns the probability density of the noncentral chi-square distribution, with df degrees of freedom and the specified noncentrality parameter, at quant.

NPDF.F. NPDF.F(quant, df1, df2, nc). Numeric. Returns the probability density of the noncentral F distribution, with degrees of freedom df1 and df2 and noncentrality nc, at quant.

NPDF.T. NPDF.T(quant, df, nc). Numeric. Returns the probability density of the noncentral Student's t distribution, with the specified degrees of freedom df and noncentrality nc, at quant.

MATRIX tail probability functions

The following functions give the probability that a random variable with the specified distribution will be greater than *quant*, the first argument. Subsequent arguments are the parameters of the distribution. Note the period in each function name.

SIG.CHISQ. SIG.CHISQ(quant, df). Numeric. Returns the cumulative probability that a value from the chi-square distribution, with df degrees of freedom, will be greater than quant

SIG.F. SIG.F(quant, df1, df2). Numeric. Returns the cumulative probability that a value from the F distribution, with degrees of freedom df1 and df2, will be greater than quant.

MATRIX cumulative distribution functions

The following functions give the probability that a random variable with the specified distribution will be less than *quant*, the first argument. All functions accept matrix, vector, or scalar as the first argument. All functions except **`CDF.BVNOR`** accept only scalar as the second argument. The result has the same dimension as the first argument. Note the period in each function name.

CDF.BERNOULLI. CDF.BERNOULLI(quant, prob). Numeric. Returns the cumulative probability that a value from the Bernoulli distribution, with the given probability parameter, will be less than or equal to quant.

CDF.BETA. CDF.BETA(quant, shape1, shape2). Numeric. Returns the cumulative probability that a value from the Beta distribution, with the given shape parameters, will be less than quant.

CDF.BINOM. CDF.BINOM(quant, n, prob). Numeric. Returns the cumulative probability that the number of successes in n trials, with probability prob of success in each, will be less than or equal to quant. When n is 1, this is the same as CDF.BERNOULLI.

CDF.BVNOR. CDF.BVNOR(quant1, quant2, corr). Numeric. Returns the cumulative probability that a value from the standard bivariate normal distribution, with the given correlation parameter, will be less than quant1 and quant2.

Both the first and second arguments can be matrices or vectors, in which case they should have the same dimensions and the result will have the same dimension as either of these arguments. It is possible that either first or second argument is a matrix or vector, and the other argument is scalar. In this case, the result has the same dimension as the matrix/vector. It is also possible that both the first and second arguments are scalars (in which case, the result will be scalar). The function's third argument should always be scalar.

CDF.CAUCHY. CDF.CAUCHY(quant, loc, scale). Numeric. Returns the cumulative probability that a value from the Cauchy distribution, with the given location and scale parameters, will be less than quant.

CDF.CHISQ. CDF.CHISQ(quant, df). Numeric and same as CHICDF. Chi-square cumulative distribution function of elements. Takes two arguments, a matrix of chi-square values and a scalar giving the degrees of freedom (which must be positive). Returns a matrix having the same dimensions as the first argument, containing the values of the cumulative chi-square distribution function for each of its elements. If an element of the first argument is *x* and the second argument is *S*, the corresponding element of the result is a number between 0 and 1, giving the proportion of a chi-square distribution with *S* degrees of freedom that is less than *x*. If *x* is not positive, the result is 0.

CDF.EXP. CDF.EXP(quant, scale). Numeric. Returns the cumulative probability that a value from the exponential distribution, with the given scale parameter, will be less than quant.

CDF.F. CDF.F(M,S1,S2). Numeric and same as FCDF. *Cumulative* F *distribution function of elements.* Takes three arguments, a matrix of *F* values and two scalars giving the degrees of freedom (which must be positive). Returns a matrix having the same dimensions as the first argument *M*, containing the values of the cumulative *F* distribution function for each of its elements. If an element of the first argument is *x* and the second and third arguments are *S1* and *S2*, the corresponding element of the result is a number between 0 and 1, giving the proportion of an *F* distribution with *S1* and *S2* degrees of freedom that is less than *x*. If *x* is not positive, the result is 0.

CDF.GAMMA. CDF.GAMMA(quant, shape, scale). Numeric. Returns the cumulative probability that a value from the Gamma distribution, with the given shape and scale parameters, will be less than quant.

CDF.GEOM. CDF.GEOM(quant, prob). Numeric. Returns the cumulative probability that the number of trials to obtain a success, when the probability of success is given by prob, will be less than or equal to quant.

CDF.HALFNRM. CDF.HALFNRM(quant, mean, stddev). Numeric. Returns the cumulative probability that a value from the half normal distribution, with specified mean and standard deviation, will be less than quant.

CDF.HYPER. CDF.HYPER(quant, total, sample, hits). Numeric. Returns the cumulative probability that the number of objects with a specified characteristic, when sample objects are randomly selected from a universe of size total in which hits have the specified characteristic, will be less than or equal to quant.

CDF.IGAUSS. CDF.IGAUSS(quant, loc, scale). Numeric. Returns the cumulative probability that a value from the inverse Gaussian distribution, with the given location and scale parameters, will be less than quant.

CDF.LAPLACE. CDF.LAPLACE(quant, mean, scale). Numeric. Returns the cumulative probability that a value from the Laplace distribution, with the specified mean and scale parameters, will be less than quant.

CDF.LNORMAL. CDF.LNORMAL(quant, a, b). Numeric. Returns the cumulative probability that a value from the log-normal distribution, with the specified parameters, will be less than quant.

CDF.LOGISTIC. CDF.LOGISTIC(quant, mean, scale). Numeric. Returns the cumulative probability that a value from the logistic distribution, with the specified mean and scale parameters, will be less than quant.

CDF.NEGBIN. CDF.NEGBIN(quant, thresh, prob). Numeric. Returns the cumulative probability that the number of trials to obtain a success, when the threshold parameter is thresh and the probability of success is given by prob, will be less than or equal to quant.

CDF.NORMAL. CDF.NORMAL(quant, mean, stddev). Numeric. Returns the cumulative probability that a value from the normal distribution, with specified mean and standard deviation, will be less than quant.

CDF.PARETO. CDF.PARETO(quant, threshold, shape). Numeric. Returns the cumulative probability that a value from the Pareto distribution, with the specified threshold and shape parameters, will be less than quant.

CDF.POISSON. CDF.POISSON(quant, mean). Numeric. Returns the cumulative probability that a value from the Poisson distribution, with the specified mean or rate parameter, will be less than or equal to quant.

CDF.SMOD. CDF.SMOD(quant, a, b). Numeric. Returns the cumulative probability that a value from the Studentized maximum modulus, with the specified parameters, will be less than quant.

CDF.SRANGE. CDF.SRANGE(quant, a, b). Numeric. Returns the cumulative probability that a value from the Studentized range statistic, with the specified parameters, will be less than quant.

CDF.T. CDF.T(M,S). Numeric and same as TCDF. *Cumulative* t *distribution function of elements.* Takes two arguments, a matrix of *t* values and a scalar giving the degrees of freedom (which must be positive). Returns a matrix having the same dimensions as *M*, containing the values of the cumulative *t* distribution function for each of its elements. If an element of the first argument is *x* and the second argument is *S*, then the corresponding element of the result is a number between 0 and 1, giving the proportion of a *t* distribution with *S* degrees of freedom that is less than *x*.

CDF.UNIFORM. CDF.UNIFORM(quant, min, max). Numeric. Returns the cumulative probability that a value from the uniform distribution, with the specified minimum and maximum, will be less than quant.

CDF.WEIBULL. CDF.WEIBULL(quant, a, b). Numeric. Returns the cumulative probability that a value from the Weibull distribution, with the specified parameters, will be less than quant.

CDFNORM(M). *Standard normal cumulative distribution function of elements.* Takes a single argument. Returns a matrix having the same dimensions as the argument, containing the values of the cumulative normal distribution function for each of its elements. If an element of the argument is *x*, the corresponding element of the result is a number between 0 and 1, giving the proportion of a normal distribution that is less than *x*. For example, `CDFNORM({-1.96,0,1.96})` results in, approximately, {.025,.5,.975}.

Note: CDFNORM(M) is a special case of CDF.NORMAL(x, mean, stddev), where mean = 0, stddev = 1.

CHICDF. CHICDF(M,S). Numeric and same as CDF.CHISQ. Chi-square cumulative distribution function of elements. Takes two arguments, a matrix of chi-square values and a scalar giving the degrees of freedom (which must be positive). Returns a matrix having the same dimensions as the first argument, containing the values of the cumulative chi-square distribution function for each of its elements. If an element of the first argument is *x* and the second argument is *S*, the corresponding element of the result is a number between 0 and 1, giving the proportion of a chi-square distribution with *S* degrees of freedom that is less than *x*. If *x* is not positive, the result is 0.

FCDF. FCDF(M,S1,S2). Numeric and same as CDF.F. *Cumulative* F *distribution function of elements.* Takes three arguments, a matrix of *F* values and two scalars giving the degrees of freedom (which must be positive). Returns a matrix having the same dimensions as the first argument *M*, containing the values of the cumulative *F* distribution function for each of its elements. If an element of the first argument is *x* and the second and third arguments are *S1* and *S2*, the corresponding element of the result is a number between 0 and 1, giving the proportion of an *F* distribution with *S1* and *S2* degrees of freedom that is less than *x*. If *x* is not positive, the result is 0.

NCDF.BETA. NCDF.BETA(quant, shape1, shape2, nc). Numeric. Returns the cumulative probability that a value from the noncentral Beta distribution, with the given shape and noncentrality parameters, will be less than quant.

NCDF.CHISQ. NCDF.CHISQ(quant, df, nc). Numeric. Returns the cumulative probability that a value from the noncentral chi-square distribution, with df degrees of freedom and the specified noncentrality parameter, will be less than quant.

NCDF.F. NCDF.F(quant, df1, df2, nc). Numeric. Returns the cumulative probability that a value from the noncentral F distribution, with degrees of freedom df1 and df2, and noncentrality nc, will be less than quant.

NCDF.T. NCDF.T(quant, df, nc). Numeric. Returns the cumulative probability that a value from the noncentral Student's t distribution, with the specified degrees of freedom df and noncentrality nc, will be less than quant.

TCDF. TCDF(M,S). Numeric and same as CDF.T. *Cumulative* t *distribution function of elements.* Takes two arguments, a matrix of *t* values and a scalar giving the degrees of freedom (which must be positive). Returns a matrix having the same dimensions as *M*, containing the values of the cumulative *t* distribution function for each of its elements. If an element of the first argument is *x* and the second argument is *S*, then the corresponding element of the result is a number between 0 and 1, giving the proportion of a *t* distribution with *S* degrees of freedom that is less than *x*.

MATRIX inverse distribution functions

The following functions give the value in a specified distribution having a cumulative probability equal to *prob*, the first argument. Subsequent arguments are the parameters of the distribution. Note the period in each function name.

IDF.BETA. IDF.BETA(prob, shape1, shape2). Numeric. Returns the value from the Beta distribution, with the given shape parameters, for which the cumulative probability is prob.

IDF.CAUCHY. IDF.CAUCHY(prob, loc, scale). Numeric. Returns the value from the Cauchy distribution, with the given location and scale parameters, for which the cumulative probability is prob.

IDF.CHISQ. IDF.CHISQ(prob, df). Numeric. Returns the value from the chi-square distribution, with the specified degrees of freedom df, for which the cumulative probability is prob. For example, the chi-square value that is significant at the 0.05 level with 3 degrees of freedom is IDF.CHISQ(0.95,3).

IDF.EXP. IDF.EXP(p, scale). Numeric. Returns the value of an exponentially decaying variable, with rate of decay scale, for which the cumulative probability is p.

IDF.F. IDF.F(prob, df1, df2). Numeric. Returns the value from the F distribution, with the specified degrees of freedom, for which the cumulative probability is prob. For example, the F value that is significant at the 0.05 level with 3 and 100 degrees of freedom is IDF.F(0.95,3,100).

IDF.GAMMA. IDF.GAMMA(prob, shape, scale). Numeric. Returns the value from the Gamma distribution, with the specified shape and scale parameters, for which the cumulative probability is prob.

IDF.HALFNRM. IDF.HALFNRM(prob, mean, stddev). Numeric. Returns the value from the half normal distribution, with the specified mean and standard deviation, for which the cumulative probability is prob.

IDF.IGAUSS. IDF.IGAUSS(prob, loc, scale). Numeric. Returns the value from the inverse Gaussian distribution, with the given location and scale parameters, for which the cumulative probability is prob.

IDF.LAPLACE. IDF.LAPLACE(prob, mean, scale). Numeric. Returns the value from the Laplace distribution, with the specified mean and scale parameters, for which the cumulative probability is prob.

IDF.LNORMAL. IDF.LNORMAL(prob, a, b). Numeric. Returns the value from the log-normal distribution, with specified parameters, for which the cumulative probability is prob.

IDF.LOGISTIC. IDF.LOGISTIC(prob, mean, scale). Numeric. Returns the value from the logistic distribution, with specified mean and scale parameters, for which the cumulative probability is prob.

IDF.NORMAL. IDF.NORMAL(prob, mean, stddev). Numeric. Returns the value from the normal distribution, with specified mean and standard deviation, for which the cumulative probability is prob.

IDF.PARETO. IDF.PARETO(prob, threshold, shape). Numeric. Returns the value from the Pareto distribution, with specified threshold and scale parameters, for which the cumulative probability is prob.

IDF.SMOD. IDF.SMOD(prob, a, b). Numeric. Returns the value from the Studentized maximum modulus, with the specified parameters, for which the cumulative probability is prob.

IDF.SRANGE. IDF.SRANGE(prob, a, b). Numeric. Returns the value from the Studentized range statistic, with the specified parameters, for which the cumulative probability is prob.

IDF.T. IDF.T(prob, df). Numeric. Returns the value from Student's t distribution, with specified degrees of freedom df, for which the cumulative probability is prob.

IDF.UNIFORM. IDF.UNIFORM(prob, min, max). Numeric. Returns the value from the uniform distribution between min and max for which the cumulative probability is prob.

IDF.WEIBULL. IDF.WEIBULL(prob, a, b). Numeric. Returns the value from the Weibull distribution, with specified parameters, for which the cumulative probability is prob.

PROBIT. PROBIT(prob). Numeric. Returns the value in a standard normal distribution having a cumulative probability equal to prob. The argument prob is a probability greater than 0 and less than 1.

Note: PROBIT(prob) is the inverse of CDFNORM(M) and is a special case of IDF.NORMAL(prob, mean, stddev), where mean = 0, stddev = 1.

MATRIX random variable functions

The following functions give a random variate from a specified distribution. The arguments are the parameters of the distribution. The first and second variables You can repeat the sequence of pseudorandom numbers by setting a seed in the Preferences dialog box before each sequence. Note the period in each function name.

NORMAL. NORMAL(stddev). Numeric. Returns a normally distributed pseudorandom number from a distribution with mean 0 and standard deviation stddev, which must be a positive number. You can repeat the sequence of pseudorandom numbers by setting a seed in the Random Number Seed dialog box before each sequence.

RV.BERNOULLI. RV.BERNOULLI(prob). Numeric. Returns a random value from a Bernoulli distribution with the specified probability parameter prob.

RV.BETA. RV.BETA(shape1, shape2). Numeric. Returns a random value from a Beta distribution with specified shape parameters.

RV.BINOM. RV.BINOM(n, prob). Numeric. Returns a random value from a binomial distribution with specified number of trials and probability parameter.

RV.CAUCHY. RV.CAUCHY(loc, scale). Numeric. Returns a random value from a Cauchy distribution with specified location and scale parameters.

RV.CHISQ. RV.CHISQ(df). Numeric. Returns a random value from a chi-square distribution with specified degrees of freedom df.

RV.EXP. RV.EXP(scale). Numeric. Returns a random value from an exponential distribution with specified scale parameter.

RV.F. RV.F(df1, df2). Numeric. Returns a random value from an F distribution with specified degrees of freedom, df1 and df2.

RV.GAMMA. RV.GAMMA(shape, scale). Numeric. Returns a random value from a Gamma distribution with specified shape and scale parameters.

RV.GEOM. RV.GEOM(prob). Numeric. Returns a random value from a geometric distribution with specified probability parameter.

RV.HALFNRM. RV.HALFNRM(mean, stddev). Numeric. Returns a random value from a half normal distribution with the specified mean and standard deviation.

RV.HYPER. RV.HYPER(total, sample, hits). Numeric. Returns a random value from a hypergeometric distribution with specified parameters.

RV.IGAUSS. RV.IGAUSS(loc, scale). Numeric. Returns a random value from an inverse Gaussian distribution with the specified location and scale parameters.

RV.LAPLACE. RV.LAPLACE(mean, scale). Numeric. Returns a random value from a Laplace distribution with specified mean and scale parameters.

RV.LOGISTIC. RV.LOGISTIC(mean, scale). Numeric. Returns a random value from a logistic distribution with specified mean and scale parameters.

RV.LNORMAL. RV.LNORMAL(a, b). Numeric. Returns a random value from a log-normal distribution with specified parameters.

RV.NEGBIN. RV.NEGBIN(threshold, prob). Numeric. Returns a random value from a negative binomial distribution with specified threshold and probability parameters.

RV.NORMAL. RV.NORMAL(mean, stddev). Numeric. Returns a random value from a normal distribution with specified mean and standard deviation.

RV.PARETO. RV.PARETO(threshold, shape). Numeric. Returns a random value from a Pareto distribution with specified threshold and shape parameters.

RV.POISSON. RV.POISSON(mean). Numeric. Returns a random value from a Poisson distribution with specified mean/rate parameter.

RV.T. RV.T(df). Numeric. Returns a random value from a Student's t distribution with specified degrees of freedom df.

RV.UNIFORM. RV.UNIFORM(min, max). Numeric. Returns a random value from a uniform distribution with specified minimum and maximum. See also the UNIFORM function.

WEIBULL. RV.WEIBULL(a, b). Numeric. Returns a random value from a Weibull distribution with specified parameters.

UNIFORM. UNIFORM(max). Numeric. Returns a uniformly distributed pseudorandom number between 0 and the argument max, which must be numeric (but can be negative). You can repeat the sequence of pseudorandom numbers by setting the same Random Number Seed (available in the Transform menu) before each sequence.

CALL Statement

Closely related to the matrix functions are the matrix procedures, which are invoked with the `CALL` statement. Procedures, similarly to functions, accept arguments enclosed in parentheses and separated by commas. They return their result in one or more of the arguments as noted in the individual descriptions below. They are implemented as procedures rather than as functions so that they can return more than one value or (in the case of `SETDIAG`) modify a matrix without making a copy of it.

EIGEN(M,var1,var2). *Eigenvectors and eigenvalues of a symmetric matrix.* Takes three arguments: a symmetric matrix and two valid variable names to which the results are assigned. If *M* is a symmetric matrix, the statement `CALL EIGEN(M, A, B)` will assign to *A* a matrix having the same dimensions as *M*, containing the eigenvectors of *M* as its columns, and will assign to *B* a column vector having as many rows as *M*, containing the eigenvalues of *M* in descending numerical order. The eigenvectors in *A* are ordered to correspond with the eigenvalues in *B*; thus, the first column corresponds to the largest eigenvalue, the second to the second largest, and so on.

SETDIAG(M,V). *Set the main diagonal of a matrix.* Takes two arguments, a matrix and a vector. Elements on the main diagonal of *M* are set equal to the corresponding elements of *V*. If *V* is a scalar, all the diagonal elements are set equal to that scalar. Otherwise, if *V* has fewer elements than the main diagonal of *M*, remaining elements on the main diagonal are unchanged. If *V* has more elements than are needed, the extra elements are not used. See also the `MDIAG` matrix function.

SVD(M,var1,var2,var3). *Singular value decomposition of a matrix.* Takes four arguments: a matrix and three valid variable names to which the results are assigned. If *M* is a matrix, the statement `CALL SVD(M,U,Q,V)` will assign to *Q* a diagonal matrix of the same dimensions as *M*, and to *U* and *V* unitary matrices (matrices whose inverses equal their transposes) of appropriate dimensions, such that *M=U*Q** T(*V*), where T is the transpose function defined above. The singular values of *M* are in the main diagonal of *Q*.

PRINT Statement

The `PRINT` statement displays matrices or matrix expressions. Its syntax is as follows:

```
PRINT [matrix expression]
       [/FORMAT="format descriptor"]
       [/TITLE="title"]
       [/SPACE={NEWPAGE}]
               {n      }
       [{/RLABELS=list of quoted names}]
        {/RNAMES=vector of names      }

       [{/CLABELS=list of quoted names}]
        {/CNAMES=vector of names      }
```

Notes

- The MDISPLAY global setting specifies how matrices are displayed. By default, IBM SPSS Statistics displays matrices to TEXT. The TABLES setting uses pivot tables to display matrices. When the **MDISPLAY** global setting is used, the PRINT statement uses either TEXT or TABLES mode exclusively (whichever is specified for **MDISPLAY**).
- When the PRINT statement is used in TABLES mode, it supports long row and column names when they are explicitly specified in either the /RLABELS or /CLABELS subcommands. When a PRINT statement is used in TEXT mode, row and column names are truncated to 8 bytes on any PRINT subcommand in which they are specified. When row and column names are not specified, default names `row1`, `row2`, ... and `col1`, `col2`, ... are used.
- When the PRINT statement is used in TABLES mode, the scaling factor is shown in the output annotation.

Matrix Expression

Matrix expression is a single matrix variable name or an expression that evaluates to a matrix. PRINT displays the specified matrix.

- The matrix specification must precede any other specifications on the PRINT statement. If no matrix is specified, no data will be displayed, but the TITLE and SPACE specifications will be honored.
- You can specify a matrix name, a matrix raised to a power, or a matrix function (with its arguments in parentheses) by itself, but you must enclose other matrix expressions in parentheses. For example, `PRINT A`, `PRINT INV(A)`, and `PRINT B**DET(T(C)*D)` are all legal, but `PRINT A+B` is not. You must specify `PRINT (A+B)`.
- Constant expressions are allowed.
- A matrix program can consist entirely of PRINT statements, without defining any matrix variables.

FORMAT Keyword

FORMAT specifies a single format descriptor for display of the matrix data.

- All matrix elements are displayed with the same format.
- You can use any printable numeric format (for numeric matrices) or string format (for string matrices) as defined in FORMATS.
- The matrix processor will choose a suitable numeric format if you omit the FORMAT specification, but a string format such as A8 is essential when displaying a matrix containing string data.
- String values exceeding the width of a string format are truncated.
- See "Scaling Factor in Displays " on page 69 for default formatting of matrices containing large or small values.

TITLE Keyword

TITLE specifies a title for the matrix displayed. The title must be enclosed in quotes. If it exceeds the maximum display width, it is truncated. The slash preceding TITLE is required, even if it is the only specification on the PRINT statement. If you omit the TITLE specification, the matrix name or expression from the PRINT statement is used as a default title.

SPACE Keyword

SPACE controls output spacing before printing the title and the matrix. You can specify either a positive number or the keyword NEWPAGE. The slash preceding SPACE is required, even if it is the only specification on the PRINT statement.

NEWPAGE. *Start a new page before printing the title.*

n. *Skip* n *lines before displaying the title.*

Note: The /SPACE subcommand is ignored when MDISPLAY is in TABLES mode. However, /SPACE is not ignored when only TITLE and no matrix is specified. In this case, the title is printed as in TEXT mode.

RLABELS Keyword

`RLABELS` allows you to supply row labels for the matrix.

- The labels must be separated by commas.
- Enclose individual labels in quotes if they contain embedded commas or if you want to preserve lowercase letters. Otherwise, quotes are optional.
- If too many names are supplied, the extras are ignored. If not enough names are supplied, the last rows remain unlabeled.
- When the `PRINT` statement is used in `TABLES` mode, the `/RLABELS` subcommand supports long row names (up to 64 bytes). When the `PRINT` statement is used in TEXT mode, long row names are truncated to 8 bytes.

RNAMES Keyword

`RNAMES` allows you to supply the name of a vector or a vector expression containing row labels for the matrix.

- Either a row vector or a column vector can be used, but the vector must contain string data.
- If too many names are supplied, the extras are ignored. If not enough names are supplied, the last rows remain unlabeled.
- Row names that are referenced through a vector in `/RNAMES` are truncated to 8 bytes. When row names are not specified, default names `row1`, `row2`,... are used.

CLABELS Keyword

`CLABELS` allows you to supply column labels for the matrix.

- The labels must be separated by commas.
- Enclose individual labels in quotes if they contain embedded commas or if you want to preserve lowercase letters. Otherwise, quotes are optional.
- If too many names are supplied, the extras are ignored. If not enough names are supplied, the last columns remain unlabeled.
- When the `PRINT` statement is used in `TABLES` mode, the `/CLABELS` subcommand supports long column names (up to 64 bytes). When the `PRINT` statement is used in TEXT mode, long column names are truncated to 8 bytes.

CNAMES Keyword

`CNAMES` allows you to supply the name of a vector or a vector expression containing column labels for the matrix.

- Either a row vector or a column vector can be used, but the vector must contain string data.
- If too many names are supplied, the extras are ignored. If not enough names are supplied, the last columns remain unlabeled.
- Column names that are referenced through a vector in `/CNAMES` are truncated to 8 bytes. When column names are not specified, default names `col1`, `col2`,... are used.

Scaling Factor in Displays

When a matrix contains very large or very small numbers, it may be necessary to use scientific notation to display the data. If you do not specify a display format, the matrix processor chooses a power-of-10 multiplier that will allow the largest value to be displayed, and it displays this multiplier on a heading line before the data. The multiplier is not displayed for each element in the matrix. The displayed values, multiplied by the power of 10 that is indicated in the heading, equal the actual values (possibly rounded).

- Values that are very small, relative to the multiplier, are displayed as `0`.
- If you explicitly specify a scientific-notation format (`Ew.d`), each matrix element is displayed using that format. This permits you to display very large and very small numbers in the same matrix without losing precision.

Example

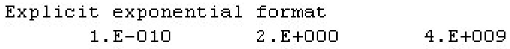
```
COMPUTE M = {.0000000001357, 2.468, 3690000000}.
PRINT M /TITLE "Default format".
PRINT M /FORMAT "E13" /TITLE "Explicit exponential format".
```

- The first PRINT subcommand uses the default format with 10^9 as the multiplier for each element of the matrix. This results in the following output:

```
Default format
  10 ** 9   X
    .000000000    .000000002   3.690000000
```

Note that the first element is displayed as 0 and the second is rounded to one significant digit.

- An explicitly specified exponential format on the second PRINT subcommand allows each element to be displayed with full precision, as the following output shows:



```
Explicit exponential format
       1.E-010       2.E+000       4.E+009
```

Matrix Control Structures

The matrix language includes two structures that allow you to alter the flow of control within a matrix program.

- The `DO IF` statement tests a logical expression to determine whether one or more subsequent matrix statements should be executed.
- The `LOOP` statement defines the beginning of a block of matrix statements that should be executed repeatedly until a termination criterion is satisfied or a BREAK statement is executed.

These statements closely resemble the `DO IF` and `LOOP` commands in the IBM SPSS Statistics transformation language. In particular, these structures can be nested within one another as deeply as the available memory allows.

DO IF Structures

A `DO IF` structure in a matrix program affects the flow of control exactly as the analogous commands affect a transformation program, except that missing-value considerations do not arise in a matrix program. The syntax of the `DO IF` structure is as follows:

```
DO IF [(]logical expression[)]
  matrix statements
[ELSE IF [(]logical expression[)]]
  matrix statements
[ELSE IF...]
  .
  .
  .
[ELSE]
 matrix statements
END IF.
```

- The `DO IF` statement marks the beginning of the structure, and the `END IF` statement marks its end.
- The `ELSE IF` statement is optional and can be repeated as many times as desired within the structure.
- The ELSE statement is optional. It can be used only once and must follow any `ELSE IF` statements.
- The `END IF` statement must follow any `ELSE IF` and ELSE statements.
- The `DO IF` and `ELSE IF` statements must contain a logical expression, normally one involving the relational operators EQ, GT, and so on. However, the matrix language allows any expression that evaluates to a scalar to be used as the logical expression. Scalars greater than 0 are considered true, and scalars less than or equal to 0 are considered false.

A `DO IF` structure affects the flow of control within a matrix program as follows:

- If the logical expression on the `DO IF` statement is true, the statements immediately following the `DO IF` are executed up to the next `ELSE IF` or `ELSE` in the structure. Control then passes to the first statement following the `END IF` for that structure.
- If the expression on the `DO IF` statement is false, control passes to the first `ELSE IF`, where the logical expression is evaluated. If this expression is true, statements following the `ELSE IF` are executed up to the next `ELSE IF` or `ELSE` statement, and control passes to the first statement following the `END IF` for that structure.
- If the expressions on the `DO IF` and the first `ELSE IF` statements are both false, control passes to the next `ELSE IF`, where that logical expression is evaluated. If none of the expressions is true on any of the `ELSE IF` statements, statements following the `ELSE` statement are executed up to the `END IF` statement, and control falls out of the structure.
- If none of the expressions on the `DO IF` statement or the `ELSE IF` statements is true and there is no `ELSE` statement, control passes to the first statement following the `END IF` for that structure.

LOOP Structures

A `LOOP` structure in a matrix program affects the flow of control exactly as the analogous commands affect transformation program, except that missing-value considerations do not arise in a matrix program. Its syntax is as follows:

```
LOOP [varname=n TO m [BY k]] [IF [(]logical expression[)]]
matrix statements
[BREAK]
matrix statements
END LOOP [IF [(]logical expression[)]]
```

The matrix statements specified between `LOOP` and `END LOOP` are executed repeatedly until one of the following conditions is met:

- A logical expression on the `IF` clause of the `LOOP` statement is evaluated as false.
- An index variable used on the `LOOP` statement passes beyond its terminal value.
- A logical expression on the `IF` clause of the `END LOOP` statement is evaluated as true.
- A `BREAK` statement is executed within the loop structure (but outside of any nested loop structures).

Note: Unlike the `LOOP` command (outside the matrix language), the index value of a matrix `LOOP` structure does not override the maximum number of loops controlled by `SET MXLOOPS`. You must explicitly set the `MXLOOPS` value to a value high enough to accommodate the index value. See the topic MXLOOPS Subcommand for more information.

Index Clause on the LOOP Statement

An index clause on a `LOOP` statement creates an index variable whose name is specified immediately after the keyword `LOOP`. The variable is assigned an initial value of *n*. Each time through the loop, the variable is tested against the terminal value *m* and incremented by the increment value *k* if *k* is specified or by 1 if *k* is not specified. When the index variable is greater than *m* for positive increments or less than *m* for negative increments, control passes to the statement after the `END LOOP` statement.

- Both the index clause and the `IF` clause are optional. If both are present, the index clause must appear first.
- The index variable must be scalar with a valid matrix variable name.
- The initial value, *n*, the terminal value, *m*, and the increment, *k* (if present), must be scalars or matrix expressions evaluating to scalars. Non-integer values are truncated to integers before use.
- If the keyword `BY` and the increment *k* are absent, an increment of 1 is used.

IF Clause on the LOOP Statement

The logical expression is evaluated before each iteration of the loop structure. If it is false, the loop terminates and control passes to the statement after `END LOOP`.

- The `IF` clause is optional. If both the index clause and the `IF` clause are present, the index clause must appear first.
- As in the `DO IF` structure, the logical expression of the `IF` clause is evaluated as scalar, with positive values being treated as true and 0 or negative values, as false.

IF Clause on the END LOOP Statement

When an `IF` clause is present on an `END LOOP` statement, the logical expression is evaluated after each iteration of the loop structure. If it is true, the loop terminates and control passes to the statement following the `END LOOP` statement.

- The `IF` clause is optional.
- As in the `LOOP` statement, the logical expression of the `IF` clause is evaluated as scalar, with positive values being treated as true and 0 or negative values, as false.

BREAK Statement

The BREAK statement within a loop structure transfers control immediately to the statement following the (next) `END LOOP` statement. It is normally placed within a `DO IF` structure inside the `LOOP` structure to exit the loop when the specified conditions are met.

Example

```
LOOP LOCATION = 1, NROW(VEC).
+  DO IF (VEC(LOCATION) = TARGET).
+     BREAK.


+  END IF.
END LOOP.
```

- This loop searches for the (first) location of a specific value, *TARGET*, in a vector, *VEC*.
- The `DO IF` statement checks whether the vector element indexed by *LOCATION* equals the target.
- If so, the BREAK statement transfers control out of the loop, leaving *LOCATION* as the index of *TARGET* in *VEC*.

READ Statement: Reading Character Data

The READ statement reads data into a matrix or submatrix from a character-format file--that is, a file containing ordinary numbers or words in readable form. The syntax for the READ statement is:

```
READ  variable reference
  [/FILE = file reference]
   /FIELD = c1 TO c2 [BY w]
  [/SIZE = size expression]
  [/MODE = {RECTANGULAR}]
           {SYMMETRIC  }
  [/REREAD]
  [/FORMAT = format descriptor]
```

- The file can contain values in freefield or fixed-column format. The data can appear in any of the field formats supported by `DATA LIST`.
- More than one matrix can be read from a single input record by rereading the record.
- If the end of the file is encountered during a READ operation (that is, fewer values are available than the number of elements required by the specified matrix size), a warning message is displayed and the contents of the unread elements of the matrix are unpredictable.

Variable Specification

The variable reference on the READ statement is a matrix variable name, with or without indexes.

For a name without indexes:

- `READ` creates the specified matrix variable.
- The matrix need not exist when `READ` is executed.
- If the matrix already exists, it is replaced by the matrix read from the file.
- You must specify the size of the matrix using the `SIZE` specification.

For an indexed name:

- `READ` creates a submatrix from an existing matrix.
- The matrix variable named must already exist.
- You can define any submatrix with indexes; for example, `M(:,I)`. To define an entire existing matrix, specify `M(:,:)`.
- The `SIZE` specification can be omitted. If specified, its value must match the size of the specified submatrix.

FILE Specification

`FILE` designates the character file containing the data. It can be an actual filename in quotes, or a file handle defined on a `FILE HANDLE` command that precedes the matrix program.

- The filename or handle must specify an existing file containing character data, not a data file in IBM SPSS Statistics format or a specially formatted file of another kind, such as a spreadsheet file.
- The `FILE` specification is required on the first `READ` statement in a matrix program (first in order of appearance, not necessarily in order of execution). If you omit the `FILE` specification from a later `READ` statement, the statement uses the most recently named file (in order of appearance) on a `READ` statement in the same matrix program.

FIELD Specification

`FIELD` specifies the column positions of a fixed-format record where the data for matrix elements are located.

- The `FIELD` specification is required.
- `Startcol` is the number of the leftmost column of the input area.
- `Endcol` is the number of the rightmost column of the input area.
- Both `startcol` and `endcol` are required and both must be constants. For example, `FIELD = 9 TO 72` specifies that values to be read appear between columns 9 and 72 (inclusive) of each input record.
- The BY clause, if present, indicates that each value appears within a fixed set of columns on the input record; that is, one value is separated from the next by its column position rather than by a space or comma. Width is the width of the area designated for each value. For example, `FIELD = 1 TO 80 BY 10` indicates that there are eight possible values per record and that one will appear between columns 1 and 10 (inclusive), another between columns 11 and 20, and so on, up to columns 71 and 80. The BY value must evenly divide the length of the field. That is, `endcol-startcol+1` must be a multiple of the width.
- You can use the `FORMAT` specification to supply the same information as the BY clause of the `FIELD` specification. If you omit the BY clause and do not specify a format on the `FORMAT` specification, `READ` assumes that values are separated by blanks or commas within the designated field.

SIZE Specification

The `SIZE` specification is a matrix expression that, when evaluated, specifies the size of the matrix to be read.

- The expression should evaluate to a two-element row or column vector. The first element designates the number of rows in the matrix to be read; the second element gives the number of columns.
- Values of the `SIZE` specification are truncated to integers if necessary.
- The size expression may be a constant, such as `{5;5}`, or a matrix variable name, such as `MSIZE`, or any valid expression, such as `INFO(1,:)`.

- If you use a scalar as the size expression, a column vector containing that number of rows is read. Thus, `SIZE=1` reads a scalar, and `SIZE=3` reads a 3 × 1 column vector.

You must include a `SIZE` specification whenever you name an entire matrix (rather than a submatrix) on the `READ` statement. If you specify a submatrix, the `SIZE` specification is optional but, if included, must agree with the size of the specified submatrix.

MODE Specification

`MODE` specifies the format of the matrix to be read in. It can be either rectangular or symmetric. If the `MODE` specification is omitted, the default is `RECTANGULAR`.

RECTANGULAR. *Matrix is completely represented in file.* Each row begins on a new record, and all entries in that row are present on that and (possibly) succeeding records. This is the default if the `MODE` specification is omitted.

SYMMETRIC. *Elements of the matrix below the main diagonal are the same as those above it.* Only matrix elements on and below the main diagonal are read; elements above the diagonal are set equal to the corresponding symmetric elements below the diagonal. Each row is read beginning on a new record, although it may span more than one record. Only a single value is read from the first record, two values are read from the second, and so on.

- If `SYMMETRIC` is specified, the matrix processor first checks that the number of rows and the number of columns are the same. If the numbers, specified either on `SIZE` or on the variable reference, are not the same, an error message is displayed and the command is not executed.

REREAD Specification

The `REREAD` specification indicates that the current `READ` statement should begin with the last record read by a previous `READ` statement.

- `REREAD` has no further specifications.
- `REREAD` cannot be used on the first `READ` statement to read from a file.
- If you omit `REREAD`, the `READ` statement begins with the first record following the last one read by the previous `READ` statement.
- The `REREAD` specification is ignored on the first `READ` statement following a call to the `EOF` function for the same file.

FORMAT Specification

`FORMAT` specifies how the matrix processor should interpret the input data. The format descriptor can be any valid IBM SPSS Statistics data format, such as `F6`, `E12.2`, or `A6`, or it can be a type code; for example, `F`, `E`, or `A`.

- If you omit the `FORMAT` specification, the default is `F`.
- You can specify the width of fixed-size data fields with either a `FORMAT` specification or a `BY` clause on a `FIELD` specification. You can include it in both places only if you specify the same value.
- If you do not include either a `FORMAT` or a `BY` clause on `FIELD`, `READ` expects values separated by blanks or commas.
- An additional way of specifying the width is to supply a repetition factor without a width (for example, `10F`, `5COMMA`, or `3E`). The field width is then calculated by dividing the width of the whole input area on the `FIELD` specification by the repetition factor. A format with a digit for the repetition factor must be enclosed in quotes.
- Only one format can be specified. A specification such as `FORMAT='5F2.0 3F3.0 F2.0'` is invalid.

WRITE Statement: Writing Character Data

`WRITE` writes the value of a matrix expression to an external file. The syntax of the `WRITE` statement is:

```
WRITE  matrix expression
   [/OUTFILE = file reference]
    /FIELD = startcol TO endcol [BY width]
```

```
    [/MODE = {RECTANGULAR}]
             {TRIANGULAR }
    [/HOLD]
    [/FORMAT = format descriptor]
```

Matrix Expression Specification

Specify any matrix expression that evaluates to the value(s) to be written.

- The matrix specification must precede any other specifications on the `WRITE` statement.
- You can specify a matrix name, a matrix raised to a power, or a matrix function (with its arguments in parentheses) by itself, but you must enclose other matrix expressions in parentheses. For example, `WRITE A`, `WRITE INV(A)`, or `WRITE B**DET(T(C)*D)` is legal, but `WRITE A+B` is not. You must specify `WRITE (A+B).`
- Constant expressions are allowed.

OUTFILE Specification

`OUTFILE` designates the character file to which the matrix expression is to be written. The file reference can be an actual filename in quotes or a file handle defined on a `FILE HANDLE` command that precedes the matrix program. The filename or file handle must be a valid file specification.

- The `OUTFILE` specification is required on the first `WRITE` statement in a matrix program (first in order of appearance, not necessarily in order of execution).
- If you omit the `OUTFILE` specification from a later `WRITE` statement, the statement uses the most recently named file (in order of appearance) on a `WRITE` statement in the same matrix program.

FIELD Specification

`FIELD` specifies the column positions of a fixed-format record to which the data should be written.

- The `FIELD` specification is required.
- The start column, *c1*, is the number of the leftmost column of the output area.
- The end column, *c2*, is the number of the rightmost column of the output area.
- Both *c1* and *c2* are required, and both must be constants. For example, `FIELD = 9 TO 72` specifies that values should be written between columns 9 and 72 (inclusive) of each output record.
- The BY clause, if present, indicates how many characters should be allocated to the output value of a single matrix element. The value *w* is the width of the area designated for each value. For example, `FIELD = 1 TO 80 BY 10` indicates that up to eight values should be written per record, and that one should go between columns 1 and 10 (inclusive), another between columns 11 and 20, and so on up to columns 71 and 80. The value on the BY clause must evenly divide the length of the field. That is, *c2* – *c1* + 1 must be a multiple of *w*.
- You can use the `FORMAT` specification (see below) to supply the same information as the BY clause. If you omit the BY clause from the `FIELD` specification and do not specify a format on the `FORMAT` specification, `WRITE` uses freefield format, separating matrix elements by single blank spaces.

MODE Specification

`MODE` specifies the format of the matrix to be written. If `MODE` is not specified, the default is `RECTANGULAR`.

RECTANGULAR. *Write the entire matrix*. Each row starts a new record, and all of the values in that row are present in that and (possibly) subsequent records. This is the default if the `MODE` specification is omitted.

TRIANGULAR. *Write only the lower triangular entries and the main diagonal*. Each row begins a new record and may span more than one record. This mode may save file space.

- A matrix written with `MODE = TRIANGULAR` must be square, but it need not be symmetric. If it is not, values in the upper triangle are not written.
- A matrix written with `MODE = TRIANGULAR` may be read with `MODE = SYMMETRIC`.

HOLD Specification

`HOLD` causes the last line written by the current `WRITE` statement to be held so that the next `WRITE` to that file will write on the same line. Use `HOLD` to write more than one matrix on a line.

FORMAT Specification

`FORMAT` indicates how the internal (binary) values of matrix elements should be converted to character format for output.

- The format descriptor is any valid IBM SPSS Statistics data format, such as `F6`, `E12.2`, or `A6`, or it can be a format type code, such as `F`, `E`, or `A`. It specifies how the written data are encoded and, if a width is specified, how wide the fields containing the data are. (See `FORMATS` for valid formats.)
- If you omit the `FORMAT` specification, the default is `F`.
- The data field widths may be specified either here or after `BY` on the `FIELD` specification. You may specify the width in both places only if you give the same value.
- An additional way of specifying the width is to supply a repetition factor without a width (for example, `10F` or `5COMMA`). The field width is then calculated by dividing the width of the whole output area on the `FIELD` specification by the repetition factor. A format with a digit for the repetition factor must be enclosed in quotes.
- If the field width is not specified in any of these ways, then the freefield format is used--matrix values are written separated by one blank, and each value occupies as many positions as necessary to avoid the loss of precision. Each row of the matrix is written starting with a new output record.
- Only one format descriptor can be specified. Do *not* try to specify more than one format; for example, `'5F2.0 3F3.0 F2.0'` is invalid as a `FORMAT` specification on `WRITE`.

GET Statement: Reading IBM SPSS Statistics Data Files

`GET` reads matrices from an external IBM SPSS Statistics data file or from the active dataset. The syntax of `GET` is as follows:

```
GET variable reference
  [/FILE={file reference}]
         {*             }
  [/VARIABLES = variable list]
  [/NAMES = names vector]
  [/MISSING = {ACCEPT}]
              {OMIT  }
              {value }
  [/SYSMIS = {OMIT }]
             {value}
```

Variable Specification

The variable reference on the `GET` statement is a matrix variable name with or without indexes.

For a name without indexes:

- `GET` creates the specified matrix variable.
- The size of the matrix is determined by the amount of data read from the IBM SPSS Statistics data file or the active dataset.
- If the matrix already exists, it is replaced by the matrix read from the file.

For an indexed name:

- `GET` creates a submatrix from an existing matrix.
- The matrix variable named must already exist.
- You can define any submatrix with indexes; for example, `M(:,I)`. To define an entire existing matrix, specify `M(:,:)`.
- The indexes, along with the size of the existing matrix, specify completely the size of the submatrix, which must agree with the dimensions of the data read from the IBM SPSS Statistics data file.
- The specified submatrix is replaced by the matrix elements read from the IBM SPSS Statistics data file.

FILE Specification

`FILE` designates the IBM SPSS Statistics data file to be read. Use an asterisk, or simply omit the `FILE` specification, to designate the current active dataset.

- The file reference can be either a filename enclosed in quotes or a file handle defined on a `FILE HANDLE` command that precedes the matrix program.
- Both dataset references and physical file specifications are supported.
- If you omit the `FILE` specification, the active dataset is used.
- In a matrix program executed with the `INCLUDE` command, if a `SPLIT FILE` command is in effect, a `GET` statement that references the active dataset will read a single split-file group of cases. (A matrix program cannot be executed from a syntax window if a `SPLIT FILE` command is in effect.)

VARIABLES Specification

`VARIABLES` specifies a list of variables to be read from the IBM SPSS Statistics data file.

- The keyword `TO` can be used to reference consecutive variables on the data file.
- The variable list can consist of the keyword `ALL` to get all the variables in the data file. `ALL` is the default if the `VARIABLES` specification is omitted.
- Long variable names (up to 64 bytes) are supported when specified on the `/VARIABLES` subcommand (either explicitly or when using the `TO` or `ALL` keywords).
- All variables read from the data file should be numeric. If a string variable is specified, a warning message is issued and the string variable is skipped.

Example

```
GET M /VARIABLES = AGE, RESIDE, INCOME TO HEALTH.
```

- The variables *AGE*, *RESIDE*, and *INCOME TO HEALTH* from the active dataset will form the columns of the matrix *M*.

NAMES Specification

`NAMES` specifies a vector to store the variable names from the data file.

- If you omit the `NAMES` specification, the variable names are not available to the `MATRIX` procedure.
- Variable names are truncated to 8 bytes when they are referenced through a vector in the `/NAMES` subcommand.

MISSING Specification

`MISSING` specifies how missing values declared for the IBM SPSS Statistics data file should be handled.

- The `MISSING` specification is required if the data file contains missing values for any variable being read.
- If you omit the `MISSING` specification and a missing value is encountered for a variable being read, an error message is displayed and the `GET` statement is not executed.

The following keywords are available on the `MISSING` specification. There is no default.

ACCEPT. *Accept user-missing values for entry.* If the system-missing value exists for a variable to be read, you must specify `SYSMIS` to indicate how the system-missing value should be handled.

OMIT. *Skip an entire observation when a variable with a missing value is encountered.*

value. *Recode all missing values encountered (including the system-missing value) to the specified value for entry.* The replacement value can be any numeric constant.

SYSMIS Specification

`SYSMIS` specifies how system-missing values should be handled when you have specified `ACCEPT` on `MISSING`.

- The `SYSMIS` specification is ignored unless `ACCEPT` is specified on `MISSING`.
- If you specify `ACCEPT` on `MISSING` but omit the `SYSMIS` specification, and a system-missing value is encountered for a variable being read, an error message is displayed and the `GET` statement is not executed.

The following keywords are available on the `SYSMIS` specification. There is no default.

OMIT. *Skip an entire observation when a variable with a system-missing value is encountered.*

value. *Recode all system-missing values encountered to the specified value for entry.* The replacement value can be any numeric constant.

Example

```
GET SCORES
  /VARIABLES = TEST1,TEST2,TEST3
  /NAMES = VARNAMES
  /MISSING = ACCEPT
  /SYSMIS = -1.0.
```

- A matrix named *SCORES* is read from the active dataset.
- The variables *TEST1*, *TEST2*, and *TEST3* form the columns of the matrix, while the cases in the active dataset form the rows.
- A vector named *VARNAMES*, whose three elements contain the variable names *TEST1*, *TEST2*, and *TEST3*, is created.
- User-missing values defined in the active dataset are accepted into the matrix *SCORES*.
- System-missing values in the active dataset are converted to the value −1 in the matrix *SCORES*.

SAVE Statement: Writing IBM SPSS Statistics Data Files

SAVE writes matrices to an external IBM SPSS Statistics data file or to the current active dataset. The rows of the matrix expression become cases, and the columns become variables. The syntax of the SAVE statement is as follows:

```
SAVE matrix expression
  [/OUTFILE = {file reference}]
              {*             }
  [/VARIABLES = variable list]
  [/NAMES = names vector]
  [/STRINGS = variable list]
```

Matrix Expression Specification

The matrix expression following the keyword SAVE is any matrix language expression that evaluates to the value(s) to be written to a data file.

- The matrix specification must precede any other specifications on the SAVE statement.
- You can specify a matrix name, a matrix raised to a power, or a matrix function (with its arguments in parentheses) by itself, but you must enclose other matrix expressions in parentheses. For example, `SAVE A`, `SAVE INV(A)`, or `SAVE B**DET(T(C)*D)` is legal, but `SAVE A+B` is not. You must specify `SAVE (A+B)`.
- Constant expressions are allowed.

OUTFILE Specification

`OUTFILE` designates the file to which the matrix expression is to be written. It can be an actual filename in quotes or a file handle defined on a `FILE HANDLE` command that precedes the matrix program.

- To save a matrix expression as the active dataset, specify an asterisk (`*`). If there is no active dataset, one will be created; if there is one, it is replaced by the saved matrices.

- Both dataset references and physical file specifications are supported.
- The `OUTFILE` specification is required on the first SAVE statement in a matrix program (first in order of appearance, not necessarily in order of execution). If you omit the `OUTFILE` specification from a later SAVE statement, the statement uses the most recently named file (in order of appearance) on a SAVE statement in the same matrix program.
- If more than one SAVE statement writes to the active dataset in a single matrix program, the dictionary of the new active dataset is written on the basis of the information given by the first such SAVE. All of the subsequently saved matrices are appended to the new active dataset as additional cases. If the number of columns differs, an error occurs.
- When you execute a matrix program with the `INCLUDE` command, the SAVE statement creates a new IBM SPSS Statistics data file at the end of the matrix program's execution, so any attempt to GET the data file obtains the original data file, if any.
- When you execute a matrix program from a syntax window, SAVE creates a new IBM SPSS Statistics data file immediately, but the file remains open, so you cannot GET it until after the `END MATRIX` statement.

VARIABLES Specification

You can provide variable names for the IBM SPSS Statistics data file with the `VARIABLES` specification. The variable list is a list of valid variable names separated by commas.

- You can use the `TO` convention, as shown in the example below.
- You can also use the `NAMES` specification, discussed below, to provide variable names.
- Long variable names (up to 64 bytes) are supported when explicitly specified on the `/VARIABLES` subcommand.

Example

```
SAVE {A,B,X,Y} /OUTFILE=*
  /VARIABLES = A,B,X1 TO X50,Y1,Y2.
```

- The matrix expression on the SAVE statement constructs a matrix from two column vectors *A* and *B* and two matrices *X* and *Y*. All four matrix variables must have the same number of rows so that this matrix construction will be valid.
- The `VARIABLES` specification provides descriptive names so that the variable names in the new active dataset will resemble the names used in the matrix program.

NAMES Specification

As an alternative to the explicit list on the `VARIABLES` specification, you can specify a name list with a vector containing string values. The elements of this vector are used as names for the variables.

- The `NAMES` specification on SAVE is designed to complement the `NAMES` specification on the GET statement. Names extracted from IBM SPSS Statistics data files can be used in a new data file by specifying the same vector name on both `NAMES` specifications.
- If you specify both `VARIABLES` and `NAMES`, a warning message is displayed and the `VARIABLES` specification is used.
- If you omit both the `VARIABLES` and `NAMES` specifications, or if you do not specify names for all columns of the matrix, the `MATRIX` procedure creates default names. The names have the form *COLn*, where *n* is the column number.
- Variable names are truncated to 8 bytes when they are referenced through a vector in the `/NAMES` subcommand.

STRINGS Specification

The `STRINGS` specification provides the names of variables that contain short string data rather than numeric data.

- By default, all variables are assumed to be numeric.

- The variable list specification following `STRINGS` consists of a list of variable names separated by commas. The names must be among those used by `SAVE`.
- Long variable names (up to 64 bytes) are supported when explicitly specified on the `/STRINGS` subcommand

MGET Statement: Reading Matrix Data Files

`MGET` reads a matrix-format data file. `MGET` puts the data it reads into separate matrix variables. It also names these new variables automatically. The syntax of `MGET` is as follows:

```
MGET [ [/] FILE = file reference]
  [/TYPE = {COV   }]
           {CORR  }
           {MEAN  }
           {STDDEV}
           {N     }
           {COUNT }
```

- Since `MGET` assigns names to the matrices it reads, do not specify matrix names on the `MGET` statement.

FILE Specification

`FILE` designates a matrix-format data file. See MATRIX DATA for a discussion of matrix-format data files. To designate the active dataset (if it is a matrix-format data file), use an asterisk, or simply omit the `FILE` specification.

- The file reference can be either a filename enclosed in quotes or a file handle defined on a `FILE HANDLE` command that precedes the matrix program. The file reference cannot be the name of a dataset in the current session.
- Both dataset references and physical file specifications are supported.
- The same matrix-format data file can be read more than once.
- If you omit the `FILE` specification, the current active dataset is used.
- `MGET` ignores the `SPLIT FILE` command when reading the active dataset. It does honor the split-file groups that were in effect when the matrix-format data file was created.
- The maximum number of split-file groups that can be read is 99.
- The maximum number of cells that can be read is 99.

TYPE Specification

TYPE specifies the rowtype(s) to read from the matrix-format data file.

- By default, records of all rowtypes are read.
- If the matrix-format data file does not contain rows of the requested type, an error occurs.

Valid keywords on the TYPE specification are:

COV. *A matrix of covariances.*

CORR. *A matrix of correlation coefficients.*

MEAN. *A vector of means.*

STDDEV. *A vector of standard deviations.*

N. *A vector of numbers of cases.*

COUNT. *A vector of counts.*

Names of Matrix Variables from MGET

- The `MGET` statement automatically creates matrix variable names for the matrices it reads.
- All new variables created by `MGET` are reported to the user.
- If a matrix variable already exists with the same name that `MGET` chose for a new variable, the new variable is not created and a warning is issued. The `RELEASE` statement can be used to get rid of a

variable. A COMPUTE statement followed by RELEASE can be used to change the name of an existing matrix variable.

MGET constructs variable names in the following manner:

- The first two characters of the name identify the row type. If there are no cells and no split file groups, these two characters constitute the name:

  CV . A covariance matrix (rowtype COV)

  CR . A correlation matrix (rowtype CORR)

  MN . A vector of means (rowtype MEAN)

  SD . A vector of standard deviations (rowtype STDDEV)

  NC . A vector of numbers of cases (rowtype N)

  CN . A vector of counts (rowtype COUNT)
- Characters 3–5 of the variable name identify the cell number or the split-group number. Cell identifiers consist of the letter *F* and a two-digit cell number. Split-group identifiers consist of the letter *S* and a two-digit split-group number; for example, *MNF12* or *SDS22*.
- If there are both cells and split groups, characters 3–5 identify the cell and characters 6–8 identify the split group. The same convention for cell or split-file numbers is used; for example, *CRF12S21*.
- After the name is constructed as described above, any leading zeros are removed from the cell number and the split-group number; for example, *CNF2S99* or *CVF2S1*.

MSAVE Statement: Writing Matrix Data Files

The MSAVE statement writes matrix expressions to a matrix-format data file that can be used as matrix input to other procedures. (See MATRIX DATA for a discussion of matrix-format data files.) The syntax of MSAVE is as follows:

```
MSAVE matrix expression
   /TYPE = {COV   }
           {CORR  }
           {MEAN  }
           {STDDEV}
           {N     }
           {COUNT }
  [/OUTFILE = 'filespec']
  [/VARIABLES = variable list]
  [/SNAMES = variable list]
  [/SPLIT = split vector]
  [/FNAMES = variable list]
  [/FACTOR = factor vector]
```

- Only one matrix-format data file can be saved in a single matrix program.
- Each MSAVE statement writes records of a single rowtype. Therefore, several MSAVE statements will normally be required to write a complete matrix-format data file.
- Most specifications are retained from one MSAVE statement to the next so that it is not necessary to repeat the same specifications on a series of MSAVE statements. The exception is the FACTOR specification, as noted below.

Example

```
MSAVE M /TYPE=MEAN /OUTFILE=CORRMAT /VARIABLES=V1 TO V8.
MSAVE S /TYPE STDDEV.
MSAVE MAKE(1,8,24) /TYPE N.
MSAVE C /TYPE CORR.
```

- The series of MSAVE statements save the matrix variables *M*, *S*, and *C*, which contain, respectively, vectors of means and standard deviations and a matrix of correlation coefficients. The matrix-format data file thus created is suitable for use in a procedure such as FACTOR.
- The first MSAVE statement saves *M* as a vector of means. This statement specifies OUTFILE, a previously defined file handle, and VARIABLES, a list of variable names to be used in the IBM SPSS Statistics data file.

- The second MSAVE statement saves *S* as a vector of standard deviations. Note that the `OUTFILE` and `VARIABLES` specifications do not have to be repeated.
- The third MSAVE statement saves a vector of case counts. The matrix function MAKE constructs an eight-element vector with values equal to the case count (24 in this example).
- The last MSAVE statement saves *C*, an 8 × 8 matrix, as the correlation matrix.

Matrix Expression Specification

- The matrix expression must be specified first on the MSAVE statement.
- The matrix expression specification can be any matrix language expression that evaluates to the value(s) to be written to the matrix-format file.
- You can specify a matrix name, a matrix raised to a power, or a matrix function (with its arguments in parentheses) by itself, but you must enclose other matrix expressions in parentheses. For example, `MSAVE A`, `SAVE INV(A)`, or `MSAVE B**DET(T(C)*D)` is legal, but `MSAVE N * WT` is not. You must specify `MSAVE (N * WT)`.
- Constant expressions are allowed.

TYPE Specification

TYPE specifies the rowtype to write to the matrix-format data file. Only a single rowtype can be written by any one MSAVE statement.Valid keywords on the TYPE specification are:

COV. *A matrix of covariances.*

CORR. *A matrix of correlation coefficients.*

MEAN. *A vector of means.*

STDDEV. *A vector of standard deviations.*

N. *A vector of numbers of cases.*

COUNT. *A vector of counts.*

OUTFILE Specification

`OUTFILE` designates the matrix-format data file to which the matrices are to be written. It can be a filename in quotes or a file handle defined on a `FILE HANDLE` command that precedes the matrix program. The filename or handle must be a valid file specification. The file reference cannot be the name of a dataset in the current session.

- The `OUTFILE` specification is required on the first MSAVE statement in a matrix program.
- Since only one matrix-format data file can be written in a single matrix program, any `OUTFILE` specification on the second and later MSAVE statements in one matrix program must be the same as that on the first MSAVE statement.
- Both dataset references and physical file specifications are supported.

VARIABLES Specification

You can provide variable names for the matrix-format data file with the `VARIABLES` specification. The variable list is a list of valid variable names separated by commas. You can use the `TO` convention.

- The `VARIABLES` specification names only the data variables in the matrix. Split-file variables and grouping or factor variables are named on the SNAMES and FNAMES specifications.
- The names in the `VARIABLES` specification become the values of the special variable *VARNAME_* in the matrix-format data file for rowtypes of `CORR` and `COV`.
- You cannot specify the reserved names *ROWTYPE_* and *VARNAME_* on the `VARIABLES` specification.
- Long variable names (up to 64 bytes) are supported when explicitly specified on the `/VARIABLES` subcommand.
- If you omit the `VARIABLES` specification, the default names *COL1*, *COL2*, ..., etc., are used.

FACTOR Specification

To write a matrix-format data file with factor or group codes, you must use the FACTOR specification to provide a row matrix containing the values of each of the factors or group variables for the matrix expression being written by the current MSAVE statement.

- The factor vector must have the same number of columns as there are factors in the matrix data file being written. You can use a scalar when the groups are defined by a single variable. For example, FACTOR=1 indicates that the matrix data being written are for the value 1 of the factor variable.
- The values of the factor vector are written to the matrix-format data file as values of the factors in the file.
- To create a complete matrix-format data file with factors, you must execute an MSAVE statement for every combination of values of the factors or grouping variables (in other words, for every group). If split-file variables are also present, you must execute an MSAVE statement for every combination of factor codes within every combination of values of the split-file variables.

Example

```
MSAVE M11 /TYPE=MEAN /OUTFILE=CORRMAT /VARIABLES=V1 TO V8
    /FNAMES=SEX, GROUP /FACTOR={1,1}.
MSAVE S11 /TYPE STDDEV.
MSAVE MAKE(1,8,N(1,1)) /TYPE N.
MSAVE C11 /TYPE CORR.


MSAVE M12 /TYPE=MEAN /FACTOR={1,2}.
MSAVE S12 /TYPE STDDEV.
MSAVE MAKE(1,8,N(1,2)) /TYPE N.
MSAVE C12 /TYPE CORR.


MSAVE M21 /TYPE=MEAN /FACTOR={2,1}.
MSAVE S21 /TYPE STDDEV.
MSAVE MAKE(1,8,N(2,1)) /TYPE N.
MSAVE C21 /TYPE CORR.


MSAVE M22 /TYPE=MEAN /FACTOR={2,2}.
MSAVE S22 /TYPE STDDEV.
MSAVE MAKE(1,8,N(2,2)) /TYPE N.
MSAVE C22 /TYPE CORR.
```

- The first four MSAVE statements provide data for a group defined by the variables *SEX* and *GROUP*, with both factors having the value 1.
- The second, third, and fourth groups of four MSAVE statements provide the corresponding data for the other groups, in which *SEX* and *GROUP*, respectively, equal 1 and 2, 2 and 1, and 2 and 2.
- Within each group of MSAVE statements, a suitable number-of-cases vector is created with the matrix function MAKE.

FNAMES Specification

To write a matrix-format data file with factor or group codes, you can use the FNAMES specification to provide variable names for the grouping or factor variables.

- The variable list following the keyword FNAMES is a list of valid variable names, separated by commas.
- If you omit the FNAMES specification, the default names *FAC1*, *FAC2*, ..., etc., are used.

SPLIT Specification

To write a matrix-format data file with split-file groups, you must use the SPLIT specification to provide a row matrix containing the values of each of the split-file variables for the matrix expression being written by the current MSAVE statement.

- The split vector must have the same number of columns as there are split-file variables in the matrix data file being written. You can use a scalar when there is only one split-file variable. For example, SPLIT=3 indicates that the matrix data being written are for the value 3 of the split-file variable.
- The values of the split vector are written to the matrix-format data file as values of the split-file variable(s).
- To create a complete matrix-format data file with split-file variables, you must execute MSAVE statements for every combination of values of the split-file variables. (If factor variables are present,

you must execute MSAVE statements for every combination of factor codes within every combination of values of the split-file variables.)

SNAMES Specification

To write a matrix-format data file with split-file groups, you can use the SNAMES specification to provide variable names for the split-file variables.

- The variable list following the keyword SNAMES is a list of valid variable names separated by commas.
- If you omit the SNAMES specification, the default names *SPL1*, *SPL2*, ..., etc., are used.

DISPLAY Statement

`DISPLAY` provides information on the matrix variables currently defined in a matrix program and on usage of internal memory by the matrix processor. Two keywords are available on `DISPLAY`:

DICTIONARY. *Display variable name and row and column dimensions for each matrix variable currently defined.*

STATUS. *Display the status and size of internal tables.* This display is intended as a debugging aid when writing large matrix programs that approach the memory limitations of your system.

If you enter the `DISPLAY` statement with no specifications, both `DICTIONARY` and `STATUS` information is displayed.

RELEASE Statement

Use the `RELEASE` statement to release the work areas in memory assigned to matrix variables that are no longer needed.

- Specify a list of currently defined matrix variables. Variable names on the list must be separated by commas.
- `RELEASE` discards the contents of the named matrix variables. Releasing a large matrix when it is no longer needed makes memory available for additional matrix variables.
- All matrix variables are released when the `END MATRIX` statement is encountered.

Macros Using the Matrix Language

Macro expansion (see DEFINE-!ENDDEFINE) occurs before command lines are passed to the matrix processor. Therefore, previously defined macro names can be used within a matrix program. If the macro name expands to one or more valid matrix statements, the matrix processor will execute those statements. Similarly, you can define an entire matrix program, including the `MATRIX` and `END MATRIX` commands, as a macro, but you cannot define a macro within a matrix program, since `DEFINE` and `END DEFINE` are not valid matrix statements.

NONPAR CORR

`NONPAR CORR` is available in Statistics Base Edition.

```
NONPAR CORR VARIABLES= varlist [WITH varlist] [/varlist...]

 [/PRINT={TWOTAIL**}  {SIG**}  {SPEARMAN**} {FULL**}]
         {ONETAIL  }  {NOSIG}  {KENDALL   } {LOWER }
                               {BOTH      }

 [/SAMPLE]

 [/MISSING=[{PAIRWISE**} [INCLUDE]]
            {LISTWISE  }

 [/MATRIX=OUT({*                  })]
              {'savfile'|'dataset'}
```

**Default if the subcommand is omitted.

This command reads the active dataset and causes execution of any pending commands. See the topic Command Order for more information.

Example

```
NONPAR CORR
  VARIABLES=PRESTIGE SPPRES PAPRES16 DEGREE PADEG MADEG.
```

Release History

Release 27.0

- Added support for the **`FULL`** and **`LOWER`** keywords in the **`PRINT`** subcommand.

Overview

`NONPAR CORR` computes two rank-order correlation coefficients, Spearman's rho and Kendall's tau-*b*, with their significance levels. You can obtain one or both coefficients. `NONPAR CORR` automatically computes the ranks and stores the cases in memory. Therefore, memory requirements are directly proportional to the number of cases that are being analyzed.

Options

Coefficients and Significance Levels. By default, `NONPAR CORR` computes Spearman coefficients and displays the two-tailed significance level. You can request a one-tailed test, and you can display the significance level for each coefficient as an annotation by using the `PRINT` subcommand.

Random Sampling. You can request a random sample of cases by using the `SAMPLE` subcommand when there is not enough space to store all cases.

Matrix Output. You can write matrix materials to a matrix data file by using the `MATRIX` subcommand. The matrix materials include the number of cases that are used to compute each coefficient and the Spearman or Kendall coefficients for each variable. These materials can be read by other procedures.

Basic Specification

The basic specification is `VARIABLES` and a list of numeric variables. By default, Spearman correlation coefficients are calculated.

Subcommand Order

- `VARIABLES` must be specified first.
- The remaining subcommands can be used in any order.

Operations

- `NONPAR CORR` produces one or more matrices of correlation coefficients. For each coefficient, `NONPAR CORR` displays the number of used cases and the significance level.
- The number of valid cases is always displayed. Depending on the specification on the `MISSING` subcommand, the number of valid cases can be displayed for each pair or in a single annotation.
- If all cases have a missing value for a given pair of variables, or if all cases have the same value for a variable, the coefficient cannot be computed. If a correlation cannot be computed, `NONPAR CORR` displays a decimal point.
- If both Spearman and Kendall coefficients are requested, and `MATRIX` is used to write matrix materials to a matrix data file, only Spearman's coefficient will be written with the matrix materials.

Limitations

- A maximum of 25 variable lists is allowed.
- A maximum of 100 variables total per `NONPAR CORR` command is allowed.

Examples

```
NONPAR CORR VARIABLES=PRESTIGE SPPRES PAPRES16 DEGREE PADEG MADEG.
```

- By default, Spearman correlation coefficients are calculated. The number of cases upon which the correlations are based and the two-tailed significance level are displayed for each correlation.

VARIABLES Subcommand

`VARIABLES` specifies the variable list.

- All variables must be numeric.
- If keyword `WITH` is not used, `NONPAR CORR` displays the correlations of each variable with every other variable in the list.
- To obtain a rectangular matrix, specify two variable lists that are separated by keyword `WITH`. `NONPAR CORR` writes a rectangular matrix of variables in the first list correlated with variables in the second list.
- Keyword `WITH` cannot be used when the `MATRIX` subcommand is used.
- You can request more than one analysis. Use a slash to separate the specifications for each analysis.

Example

```
NONPAR CORR
  VARIABLES = PRESTIGE SPPRES PAPRES16 WITH DEGREE PADEG MADEG.
```

- The three variables that are listed before `WITH` define the rows; the three variables that are listed after `WITH` define the columns of the correlation matrix.
- Spearman's rho is displayed by default.

Example

```
NONPAR CORR VARIABLES=SPPRES PAPRES16 PRESTIGE
                     /SATCITY WITH SATHOBBY SATFAM.
```

- `NONPAR CORR` produces two Correlations tables.
- By default, Spearman's rho is displayed.

PRINT Subcommand

By default, `NONPAR CORR` displays Spearman correlation coefficients. The significance levels are displayed below the coefficients. The significance level is based on a two-tailed test. Use `PRINT` to change these defaults.

Note: The Spearman and Kendall coefficients are both based on ranks.

SPEARMAN
Spearman's rho. Only Spearman coefficients are displayed. This specification is the default.

KENDALL
Kendall's tau-b. Only Kendall coefficients are displayed.

BOTH
Kendall and Spearman coefficients. Both coefficients are displayed. If `MATRIX` is used to write the correlation matrix to a matrix data file, only Spearman coefficients are written with the matrix materials.

SIG
Do not flag significant values. `SIG` is the default.

NOSIG
Flag significant values. Values significant at the 0.05 level are flagged with a single asterisk; those that are significant at the 0.01 level are flagged with two asterisks.

TWOTAIL
Two-tailed test of significance. This test is appropriate when the direction of the relationship cannot be determined in advance, as is often the case in exploratory data analysis. This specification is the default.

`ONETAIL`

One-tailed test of significance. This test is appropriate when the direction of the relationship between a pair of variables can be specified in advance of the analysis.

`FULL`

Show the full table. The full table is presented in the output.

`LOWER`

Show only the lower triangle. Only the table's lower triangle is presented in the output. The keyword allows table output to adhere to APA style guidelines.

SAMPLE Subcommand

`NONPAR CORR` must store cases in memory to build matrices. `SAMPLE` selects a random sample of cases when computer resources are insufficient to store all cases. `SAMPLE` has no additional specifications.

MISSING Subcommand

`MISSING` controls the treatment of missing values.

- `PAIRWISE` and `LISTWISE` are alternatives. You can specify `INCLUDE` with either `PAIRWISE` or `LISTWISE`.

PAIRWISE. *Exclude missing values pairwise.* Cases with a missing value for one or both variables for a specific correlation coefficient are excluded from the computation of that coefficient. This process allows the maximum available information to be used in every calculation. This process also results in a set of coefficients based on a varying number of cases. The number is displayed for each pair. This specification is the default.

LISTWISE. *Exclude missing values listwise.* Cases with a missing value for any variable that is named in a list are excluded from the computation of all coefficients in the Correlations table. The number of used cases is displayed in a single annotation. Each variable list on a command is evaluated separately. Thus, a case that is missing for one matrix might be used in another matrix. This option decreases the amount of required memory and significantly decreases computational time.

INCLUDE. *Include user-missing values*. User-missing values are treated as valid values.

MATRIX Subcommand

`MATRIX` writes matrix materials to a matrix data file. The matrix materials always include the number of cases that are used to compute each coefficient, and the materials include either the Spearman or the Kendall correlation coefficient for each variable, whichever is requested. See the topic "Format of the Matrix Data File " on page 88 for more information.

- You cannot write both Spearman's and Kendall's coefficients to the same matrix data file. To obtain both Spearman's and Kendall's coefficients in matrix format, specify separate `NONPAR CORR` commands for each coefficient and define different matrix data files for each command.
- If `PRINT=BOTH` is in effect, `NONPAR CORR` displays a matrix in the listing file for both coefficients but writes only the Spearman coefficients to the matrix data file.
- `NONPAR CORR` cannot write matrix materials for rectangular matrices (variable lists containing keyword `WITH`). If more than one variable list is specified, only the last variable list that does not use keyword `WITH` is written to the matrix data file.
- The specification on `MATRIX` is keyword `OUT` and a quoted file specification or previously declared dataset name (`DATASET DECLARE` command), enclosed in parentheses.
- If you want to use a correlation matrix that is written by `NONPAR CORR` in another procedure, change the *ROWTYPE_* value RHO or TAUB to CORR by using the `RECODE` command.
- Any documents that are contained in the active dataset are not transferred to the matrix file.

OUT ('savfile'|'dataset'). *Write a matrix data file or dataset.* Specify either a filename, a previously declared dataset name, or an asterisk, enclosed in parentheses. Filenames should be enclosed in quotes and are stored in the working directory unless a path is included as part of the file specification. If you specify an asterisk (*), the matrix data file replaces the active dataset.

Multiple nonparametric correlation tables

```
NONPAR CORR
  VARIABLES=PRESTIGE SPPRES PAPRES16 DEGREE PADEG MADEG
  /PRESTIGE TO DEGREE /PRESTIGE WITH DEGREE
  /MATRIX OUT('/data/npmat.sav').
```

- Only the matrix for *PRESTIGE* to *DEGREE* is written to the matrix data file because it is the last variable list that does not use keyword `WITH`.

Format of the Matrix Data File

- The matrix data file has two special variables that are created by the program: *ROWTYPE_* and *VARNAME_*.
- *ROWTYPE_* is a short string variable with values N and RHO for Spearman's correlation coefficient. If you specify Kendall's coefficient, the values are N and TAUB.
- *VARNAME_* is a short string variable whose values are the names of the variables that are used to form the correlation matrix. When *ROWTYPE_* is RHO (or TAUB), *VARNAME_* gives the variable that is associated with that row of the correlation matrix.
- The remaining variables in the file are the variables that are used to form the correlation matrix.

Split Files

- When split-file processing is in effect, the first variables in the matrix data file are the split variables, followed by *ROWTYPE_*, *VARNAME_*, and the variables that are used to form the correlation matrix.
- A full set of matrix materials is written for each split-file group that is defined by the split variables.
- A split variable cannot have the same name as any other variable that is written to the matrix data file.
- If split-file processing is in effect when a matrix is written, the same split file must be in effect when that matrix is read by a procedure.

Missing Values

- With `PAIRWISE` treatment of missing values (the default), the matrix of Ns that is used to compute each coefficient is included with the matrix materials.
- With `LISTWISE` or `INCLUDE` treatments, a single `N` that is used to calculate all coefficients is included with the matrix materials.

Examples

Writing results to a matrix data file

```
GET FILE='/data/GSS80.sav'
  /KEEP PRESTIGE SPPRES PAPRES16 DEGREE PADEG MADEG.
NONPAR CORR VARIABLES=PRESTIGE TO MADEG
  /MATRIX OUT('/data/npmat.sav').
```

- `NONPAR CORR` reads data from file *GSS80.sav* and writes one set of correlation matrix materials to the file *npmat.sav*.
- The active dataset is still *GSS80.sav*. Subsequent commands are executed on file *GSS80.sav*.

Replacing the active dataset with matrix results

```
GET FILE='/data/GSS80.sav'
  /KEEP PRESTIGE SPPRES PAPRES16 DEGREE PADEG MADEG.
NONPAR CORR VARIABLES=PRESTIGE TO MADEG
  /MATRIX OUT(*).
LIST.
DISPLAY DICTIONARY.
```

- `NONPAR CORR` writes the same matrix as in the example above. However, the matrix data file replaces the active dataset. The `LIST` and `DISPLAY` commands are executed on the matrix file (not on the original active dataset *GSS80.sav*).

NPTESTS

NPTESTS is available in the Statistics Base Edition.

```
NPTESTS
  [/MISSING [SCOPE={ANALYSIS**}] [USERMISSING={EXCLUDE**}]]
                   {LISTWISE  }               {INCLUDE   }
  [/CRITERIA ALPHA={0.05**}] CILEVEL={95** }
                   {value }          {value}
         [SEED={2000000**}{integer}{RANDOM}]

  [/ONESAMPLE TEST(fieldlist)                                   ]
     [CHISQUARE(
         [EXPECTED={EQUAL**                     }]
                   {CUSTOM(FREQUENCIES=valuelist
                           CATEGORIES=valuelist) }
      )]

     [BINOMIAL([TESTVALUE={0.5**}]
                          {value}
         [SUCCESSCATEGORICAL={FIRST**               }]
                             {LIST(valuelist)       }
         [SUCCESSCONTINUOUS=CUTPOINT({MIDPOINT**})]
                                     {value     }
         [CLOPPERPEARSON] [JEFFREYS] [LIKELIHOOD]
      )]

     [KOLMOGOROV_SMIRNOV(
         [NSAMPLES={1000**}{integer}]
         [MC_CILEVEL-{99**}{value}]
         [NORMAL={SAMPLE**(SIMULATION={TRUE**}{FALSE})}
                 {CUSTOM(MEAN=value
                         SD=value   )}]
         [UNIFORM={SAMPLE**            }]
                  {CUSTOM(MIN=value
                          MAX=value   )}
         [EXPONENTIAL={SAMPLE**           }]
                      {CUSTOM(MEAN=value )}
         [POISSON=CUSTOM(MEAN=value )]
      )]

     [RUNS([GROUPCATEGORICAL={SAMPLE**                   }]
                             {LIST(valuelist)          }
           [GROUPCONTINUOUS=CUTPOINT({SAMPLEMEDIAN**})]
                                     {SAMPLEMEAN    }
                                     {value         }
      )]

     [WILCOXON(TESTVALUE=value)]

  [/INDEPENDENT TEST(fieldlist)                          ]
                GROUP(fieldname)

     [MANN_WHITNEY] [WALD_WOLFOWITZ] [KOLMOGOROV_SMIRNOV]
     [HODGES_LEHMAN]
     [MOSES([TRIMOUTLIERS={SAMPLE**}])]
                          {integer }

     [KRUSKAL_WALLIS([COMPARE={NONE      }])]
                              {PAIRWISE**}
                              {STEPWISE  }

     [MEDIAN([TESTVALUE={SAMPLE**}]  [COMPARE={NONE      }])]
                        {value   }            {PAIRWISE**}
                                              {STEPWISE  }

     [JONCKHEERE_TERPSTRA(
         [ORDER={ASCENDING**     }]  [COMPARE={NONE      }]
                {DESCENDING      }            {PAIRWISE**}
                                              {STEPWISE  }
      )]

  [/RELATED TEST(fieldlist)                      ]
     [MCNEMAR( [SUCCESS={FIRST**        }])]
                        {LIST(valuelist)}

     [WILCOXON] [SIGN] [MARGINAL_HOMOGENEITY]
     [HODGES_LEHMAN]

     [COCHRAN(
         [SUCCESS={FIRST          }]
                  {LIST(valuelist)}
         [COMPARE={NONE      }]
                  {PAIRWISE**}
                  {STEPWISE  }
      )]

     [FRIEDMAN([COMPARE={NONE      }])]
                        {PAIRWISE**}
                        {STEPWISE  }

     [KENDALL([COMPARE={NONE      }])]
                       {PAIRWISE**}
                       {STEPWISE  }
```

** Default if the subcommand or keyword is omitted.

This command reads the active dataset and causes execution of any pending commands. See the topic Command Order for more information.

Release History

Release 18

- Command introduced.

Release 22

- Added support for ordinal fields to tests that were previously restricted to continuous fields.

Release 27

- Added support for the **SEED** keyword in the **CRITERIA** subcommand.
- **ONESAMPLE** subcommand:
  - Added support for the **NSAMPLES** keyword in the **KOLMOGOROV_SMIRNOV** keyword.
  - Added support for the **MC_CILEVEL** keyword in the **KOLMOGOROV_SMIRNOV** keyword.
  - Added support for the **SIMULATION** keyword in the **KOLMOGOROV_SMIRNOV** keyword.
  - Removed the **SAMPLE** setting from the **POISSON** keyword in the **KOLMOGOROV_SMIRNOV** keyword.

Example

```
NPTESTS
  /ONESAMPLE TEST(field1 TO field10).
```

Overview

`NPTESTS` is a collection of nonparametric tests. These tests make minimal assumptions about the underlying distribution of the data.

The tests that are available in `NPTESTS` can be grouped into three broad categories based on how the data are organized.

- A one-sample test analyzes one field.
- A test for related samples compares two or more fields for the same set of cases.
- An independent-samples test analyzes one field that is grouped by categories of another field.

Options

Automatic test selection

If a test subcommand (`ONESAMPLE`, `INDEPENDENT`, or `RELATED`) is specified without any test specifications, then `NPTESTS` automatically chooses the "best" tests to perform for each field specified on the subcommand. See the individual subcommands for details.

Multiple comparisons

In addition to the "omnibus" tests for *k* independent or related samples, the samples can also be tested pairwise, either by all pairwise comparisions or through a stepwise stepdown procedure.

Missing value handling

Records with missing values can be handled listwise or analysis by analysis. User-missing values on categorical fields can be treated as valid or excluded from analysis.

Basic Specification

The basic specification is a single test subcommand (`ONESAMPLE`, `INDEPENDENT`, or `RELATED`) and a TEST keyword with a list of fields to be tested.

Syntax Rules

- At least one test subcommand must be specified; all other subcommands are optional.
- Subcommands may be specified in any order.
- Only a single instance of each subcommand is allowed.

- An error occurs if a keyword is specified more than once within a subcommand.
- Parentheses, equals signs, and slashes shown in the syntax chart are required.
- The command name, subcommand names, and keywords must be spelled in full.
- Empty subcommands are not allowed.
- Any split field defined on the `SPLIT FILE` command cannot be used on this command.
- Any field specified on the `WEIGHT` command cannot be used on this command.

Operations

Note: Since measurement level can affect the results, if any variables (fields) have an unknown measurement level, an initial data pass will be performed to determine default measurement level for any variables with an unknown measurement level. For information on the criteria used to determine default measurement level, see SET SCALEMIN.

MISSING Subcommand

The `MISSING` subcommand controls the treatment of records with missing values.

```
[/MISSING [SCOPE={ANALYSIS**}] [USERMISSING={EXCLUDE**}]]
                 {LISTWISE  }               {INCLUDE  }
```

SCOPE Keyword

The `SCOPE` keyword specifies how to determine the case basis for tests.

ANALYSIS. *Exclude records with missing values on a test-by-test basis.* Records with missing values for a field that is used for a specific test are omitted from that test. On subcommands that specify several tests, each test is evaluated separately. This setting is the default.

LISTWISE. *Exclude records with missing values listwise.* Records with missing values for any field that is named on any subcommand are excluded from all analyses.

USERMISSING Keyword

The `USERMISSING` keyword is used to control whether user-missing values for categorical fields are treated as valid values. By default, user-missing values for categorical fields are treated as invalid.

- User-missing values for continuous fields are always treated as invalid.
- System-missing values for any fields are always treated as invalid.

EXCLUDE. *User-missing values for categorical fields are treated as invalid.* This is the default.

INCLUDE. *User-missing values for categorical fields are treated as valid values.*

CRITERIA Subcommand

The `CRITERIA` subcommand controls the significance level used for tests and the confidence level used for intervals.

```
[/CRITERIA ALPHA={0.05**}] CILEVEL={95** }
                 {value }          {value}
                 [SEED={2000000**}{integer}{RANDOM}]
```

ALPHA keyword

The `ALPHA` keyword specifies the significance level for all tests specified on the `NPTESTS` command.

- Specify a numeric value between 0 and 1. 0.05 is the default.

CILEVEL keyword

The `CILEVEL` keyword specifies the confidence level for all confidence intervals produced by the `NPTESTS` command.

- Specify a numeric value between 0 and 100. 95 is the default.

SEED keyword

The `SEED` keyword resets the random seed used for the Monte Carlo sampling.

The integer in the parentheses must be a single integer between 1 and 2147483647, with 2000000 as the default setting. When **`RANDOM`** is specified, a random seed will be assigned by the procedure.

- Specify a single integer between 1 and 2147483647. 2000000 is the default.

ONESAMPLE Subcommand

The `ONESAMPLE` subcommand produces one-sample nonparametric tests. The TEST keyword is required, all other keywords are optional. If `ONESAMPLE` is specified with none of the optional keywords, the following tests are performed automatically:

- Categorical fields with two values are tested using a binomial test.
- Categorical fields with more than two values are tested using a chi-square test with equal frequencies on category values found in the sample.
- Continuous fields are tested using a Kolmogorov-Smirnov test against a normal distribution with the sample mean and standard deviation.

TEST keyword

The `TEST` keyword lists the fields that you want to test.

- Specify one or more fields. Note that certain tests are not applicable to fields of a particular measurement level; for example, the chi-square test is only performed for categorical fields. `NPTESTS` automatically determines which tests are applicable to which fields. See the individual keyword descriptions for details.

CHISQUARE keyword

```
    [CHISQUARE([EXPECTED={EQUAL**                      }])]
                         {CUSTOM(FREQUENCIES=valuelist
                                 CATEGORIES=valuelist) }
```

The `CHISQUARE` keyword produces a one-sample test that computes a chi-square statistic based on the differences between the observed and expected frequencies of categories of a field.

- A separate chi-square test is performed for each and every categorical field specified on the `TEST` keyword.
- The test specifications given on the `CHISQUARE` keyword apply to all chi-square tests performed.
- If `CHISQUARE` is specified without any keywords, equal frequencies are expected in each category.

EXPECTED = EQUAL|CUSTOM(FREQUENCIES=valuelist CATEGORIES=valuelist). *Expected frequencies.*

- The `EXPECTED` keyword defines how expected frequencies are derived. The default is `EQUAL`.
- `EQUAL` produces equal frequencies among all categories in the sample. This is the default when `CHISQUARE` is specified without any other keywords..
- `CUSTOM` allows you to specify unequal frequencies for a specified list of categories.
- On the `CATEGORIES` keyword, specify a list of string or numeric values. The values in the list do not need to be present in the sample.
- On the `FREQUENCIES` keyword, specify a value greater than 0 for each category, and in the same order as the categories, on the `CATEGORIES` keyword. Custom frequencies are treated as ratios so that, for example, `FREQUENCIES=1 2 3` is equivalent to `FREQUENCIES=10 20 30`, and both specify that 1/6 of the records are expected to fall into the first category on the `CATEGORIES` keyword, 1/3 into the second, and 1/2 into the third.

- When `CUSTOM` is specified, the number of expected frequencies must match the number of category values; otherwise the test is not performed for that field.

BINOMIAL keyword

```
[BINOMIAL([TESTVALUE={0.5**}]
                     {value}
    [SUCCESSCATEGORICAL={FIRST**                  }]
                        {LIST(valuelist)         }
    [SUCCESSCONTINUOUS=CUTPOINT({MIDPOINT**})]
                                {value     }
    [CLOPPERPEARSON] [JEFFREYS] [LIKELIHOOD]
 )]
```

The `BINOMIAL` keyword produces a one-sample test of whether the observed distribution of a dichotomous field is the same as what is expected from a specified binomial distribution. In addition, you can request confidence intervals.

- A separate binomial test is performed for each and every field specified on the `TEST` keyword.
- The test specifications given on the `BINOMIAL` keyword apply to all binomial tests performed.
- If `BINOMIAL` is specified without any keywords, each categorical field is assumed to have only two values and each continuous field is dichotomized using the average of the minimum and maximum as a cut point. The distribution of each named field is compared to a binomial distribution with *p* (the proportion of cases expected in the first category) equal to 0.5.

TESTVALUE. *Hypothesized proportion.* The `TESTVALUE` keyword specifies the expected proportion of cases in the first category. Specify a value greater than 0 and less than 1. The default is 0.5.

SUCCESSCATEGORICAL=FIRST|LIST(valuelist).

- The `SUCCESSCATEGORICAL` keyword specifies how "success", the data value(s) tested against the test value, is defined for categorical fields.
- `FIRST` performs the binomial test using the first value found in the sample to define "success". This option is only applicable to nominal or ordinal fields with only two values; all other categorical fields specified on a `ONESAMPLE` subcommand where `FIRST` is used will not be tested. This is the default.
- `LIST` performs the binomial test using the specified list of values to define "success". Specify a list of string or numeric values. The values in the list do not need to be present in the sample.

SUCCESSCONTINUOUS=CUTPOINT (MIDPOINT|value). *Define success for continuous fields.* The `SUCCESSCONTINUOUS` keyword specifies how "success", the data value(s) tested against the test value, is defined for continuous fields.

- `CUTPOINT` defines values that are equal to or less than the cut point as "success". `MIDPOINT` sets the cut point at the average of the minimum and maximum values. Alternatively, specify a value for the cut point. The default is `MIDPOINT`.

CLOPPERPEARSON. *Exact interval based on the cumulative binomial distribution.*

JEFFREYS. *Bayesian interval based on the posterior distribution of p using the Jeffreys prior.*

LIKELIHOOD. *Interval based on the likelihood function for p.*

KOLMOGOROV_SMIRNOV keyword

```
[KOLMOGOROV_SMIRNOV(
    [NSAMPLES={1000**}{integer}]
    [MC_CILEVEL-{99**}{value}]
    [NORMAL={SAMPLE**(SIMULATION={TRUE**}{FALSE})}
            {CUSTOM(MEAN=value
                    SD=value   )}]
    [UNIFORM={SAMPLE**            }]
             {CUSTOM(MIN=value
                     MAX=value   )}
    [EXPONENTIAL={SAMPLE**            }]
                 {CUSTOM(MEAN=value )}
    [POISSON=CUSTOM(MEAN=value )]
 )]
```

The `KOLMOGOROV_SMIRNOV` keyword produces a one-sample test of whether the sample cumulative distribution function for a field is homogenous with a uniform, normal, Poisson, or exponential distribution.

- A separate Kolmogorov-Smirnov test is performed for each and every continuous and ordinal field specified on the `TEST` keyword.
- The test specifications given on the `KOLMOGOROV_SMIRNOV` keyword apply to all Kolmogorov-Smirnov tests performed.
- If `KOLMOGOROV_SMIRNOV` is specified without any keywords, each field is tested against a normal distribution using its sample mean and sample standard deviation.

NSAMPLES=integer. `NSAMPLES` resets the number of replicates used by the Lilliefors test for Monte Carlo sampling.

MC_CILEVEL=value.*Monte Carlo confidence interval.* `MC_CILEVEL` resets the confidence interval level that is estimated by the Kolmogorov-Smirnov test.

NORMAL (SAMPLE (SIMULATION=boolean)|CUSTOM (MEAN=value SD=value)). *Normal distribution.* `SAMPLE` uses the observed mean and standard deviation, `SIMULATION` controls whether the Monte Carlo simulation will be used to conduct the Lilliefors test for Normal distribution when the parameters are not specified, and `CUSTOM` allows you to specify parameters.

UNIFORM (SAMPLE|CUSTOM (MIN=value MAX=value)). *Uniform distribution.* `SAMPLE` uses the observed minimum and maximum, `CUSTOM` allows you to specify values.

POISSON=CUSTOM (MEAN=value)). *Poisson distribution.* `CUSTOM` allows you to specify a mean value.

EXPONENTIAL(SAMPLE|CUSTOM (MEAN=value)). *Exponential distribution.* `SAMPLE` uses the observed mean, `CUSTOM` allows you to specify a value.

RUNS keyword

```
[RUNS([GROUPCATEGORICAL={SAMPLE**                  }]
                        {LIST(valuelist)           }
      [GROUPCONTINUOUS=CUTPOINT({SAMPLEMEDIAN**})]
                                {SAMPLEMEAN    }
                                {value         }
 )]
```

The `RUNS` keyword produces a one-sample test of whether the sequence of values of a dichotomized field is random.

- A separate runs test is performed for each and every field specified on the `TEST` keyword.
- The test specifications given on the `RUNS` keyword apply to all runs tests performed.
- If `RUNS` is specified without any keywords, each categorical field is assumed to have only two values and each continuous field is dichotomized using the sample median as a cut point.

GROUPCATEGORICAL= SAMPLE|LIST(valuelist). *Determine groups for categorical fields.* `SAMPLE` is the default.

- `SAMPLE` performs the runs test using the values found in the sample to define the groups. This option is only applicable to nominal or ordinal fields with only two values; all other categorical fields specified on a `ONESAMPLE` subcommand where `SAMPLE` is used will not be tested.
- `LIST` performs the runs test using the specified list of values to define one of the groups. All other values in the sample define the other group. The values in the list do not all need to be present in the sample, but at least one record must be in each group.

GROUPCONTINUOUS= CUTPOINT (SAMPLEMEDIAN | SAMPLEMEAN | value). *Determine groups for continuous fields.* `CUTPOINT` defines values that are equal to or less than the cut point as the first group; all other values define the other group. `SAMPLEMEDIAN` sets the cut point at the sample median. `SAMPLEMEAN` sets the cut point at the sample men. Alternatively, specify a value for the cut point. The default is `SAMPLEMEDIAN`.

WILCOXON keyword

```
[WILCOXON(TESTVALUE=value)]
```

The `WILCOXON` keyword produces a one sample test of median value of a field.

- A separate Wilcoxon test is performed for each and every continuous and ordinal field specified on the `TEST` keyword.
- The test specifications given on the `WILCOXON` keyword apply to all Wilcoxon tests performed.
- The `TESTVALUE` keyword is required.

TESTVALUE=value. *Hypothesized median.* The Wilcoxon test is performed using the specified value. The `TESTVALUE` keyword is required. There is no default.

INDEPENDENT Subcommand

The `INDEPENDENT` subcommand produces independent samples nonparametric tests. The `TEST` and `GROUP` keywords are required, all other keywords are optional. If `INDEPENDENT` is specified with none of the optional keywords, the following tests are performed automatically:

- If the grouping field has only two values, continuous fields are tested using a Mann-Whitney test.
- If the grouping field has more than two values, continuous fields are tested using a Kruskal-Wallis test.
- If there are no continuous fields on the TEST keyword, no tests are performed.

TEST Keyword

The `TEST` keyword lists the fields that you want to test.

- Specify one or more continuous fields. Independent samples tests are only applicable to continuous fields.

GROUP Keyword

The `GROUP` keyword specifies the field used to determine groups. The test is performed using the values found in the sample to define the groups. Note that certain tests are not applicable when there are more than two groups. `NPTESTS` automatically determines which tests are applicable. See the individual keyword descriptions for details.

- Specify a categorical field. A grouping field that does not have nominal or ordinal measurement level will cause an error.
- `GROUP` applies to all fields that are specified on the TEST keyword.

MANN_WHITNEY Keyword

The `MANN_WHITNEY` keyword produces an independent samples test of whether two samples are from the same population. The test statistic uses the rank of each case to test whether the groups are drawn from the same population.

- A separate test is performed for each and every continuous and ordinal field specified on the TEST keyword.
- The first value in ascending order of the grouping field defines the first group and the second value defines the second. If the grouping field has more than two values, the test is not performed.

WALD_WOLFOWITZ Keyword

The `WALD_WOLFOWITZ` keyword produces an independent samples test whether the distribution of a field is the same in two independent samples. A runs test is performed with group membership as the criterion.

- A separate test is performed for each and every continuous and ordinal field specified on the TEST keyword.
- The first value in ascending order of the grouping field defines the first group and the second value defines the second. If the grouping field has more than two values, the test is not performed.

KOLMOGOROV_SMIRNOV Keyword

The `KOLMOGOROV_SMIRNOV` keyword produces an independent samples test of whether the distribution of a field is the same in two independent samples that are defined by a grouping field. The test is sensitive to any difference in median, dispersion, skewness, and so forth, between the two distributions.

- A separate test is performed for each and every continuous and ordinal field specified on the TEST keyword.
- The first value in ascending order of the grouping field defines the first group and the second value defines the second. If the grouping field has more than two values, the test is not performed.

MOSES Keyword

```
    [MOSES([TRIMOUTLIERS={SAMPLE**}])]
                         {integer }
```

The `MOSES` keyword produces an independent samples test of whether the range of a continuous or ordinal field is the same in a control group and a comparison group.

- A separate test is performed for each and every continuous and ordinal field specified on the TEST keyword.
- The test specifications given on the `MOSES` keyword apply to all Moses tests performed.
- The first value in ascending order of the grouping field defines the control group and the second defines the comparison group. If the grouping field has more than two values, the test is not performed.

TRIMOUTLIERS= SAMPLE|integer. *Records to trim from each end of the range of the control group to remove outliers.* `SAMPLE` trims 5% of the records from each end of the range of the control group to remove outliers, with a minimum of 1 record removed from each end. Specifying an integer causes that exact number of records to be trimmed. `SAMPLE` is the default.

HODGES_LEHMAN Keyword

The `HODGES_LEHMAN` keyword produces an independent samples estimate and confidence interval for the difference in the medians of two groups.

- A separate confidence interval is computed for each and every continuous and ordinal field specified on the `TEST` keyword.
- The first value in ascending order of the grouping field defines the first group and the second value defines the second. If the grouping field has more than two values, the confidence interval is not produced.

KRUSKAL_WALLIS Keyword

```
    [KRUSKAL_WALLIS([COMPARE={PAIRWISE**}])]
                             {STEPWISE  }
                             {NONE      }
```

The `KRUSKAL_WALLIS` keyword produces an independent samples test of whether *k* independent samples that are defined by a grouping field are from the same population.

- A separate test is performed for each and every continuous and ordinal field specified on the TEST keyword.
- The test specifications given on the `KRUSKAL_WALLIS` keyword apply to all tests performed.

COMPARE= PAIRWISE | STEPWISE | NONE. *Multiple comparisons.* The `COMPARE` keyword controls how and whether multiple comparisons should be performed. `PAIRWISE` produces all pairwise multiple comparisons. `STEPWISE` produces stepwise stepdown comparisons. `NONE` turns off multiple comparisons. By default, all pairwise comparisons are produced.

MEDIAN Keyword

```
    [MEDIAN([TESTVALUE={SAMPLE**}]  [COMPARE={PAIRWISE**}])]
                       {value   }            {STEPWISE  }
                                             {NONE      }
```

The `MEDIAN` keyword produces an independent samples test of whether *k* independent samples are drawn from populations with the same median.

- A separate test is performed for each and every continuous and ordinal field specified on the TEST keyword.
- The test specifications given on the `MEDIAN` keyword apply to all tests performed.

TESTVALUE= SAMPLE|value. *Median value to use in the test.* `SAMPLE` causes the test to be performed using the sample median, calculated from all records that are included in the test. Alternatively, you can specify an exact value. `SAMPLE` is the default.

COMPARE= PAIRWISE| STEPWISE| NONE. *Multiple comparisons.* The `COMPARE` keyword controls how and whether multiple comparisons should be performed. `PAIRWISE` produces all pairwise multiple comparisons. `STEPWISE` produces stepwise stepdown comparisons. `NONE` turns off multiple comparisons. By default, all pairwise comparisons are produced.

JONCKHEERE_TERPSTRA Keyword

```
    [JONCKHEERE_TERPSTRA(
        [ORDER={ASCENDING**      }]  [COMPARE={PAIRWISE**}]
               {DESCENDING       }            {STEPWISE  }
                                              {NONE      }
     )]
```

The `JONCKHEERE-TERPSTRA` keyword produces an independent samples test of whether *k* independent samples that are defined by a grouping field are from the same population. This test is particularly powerful when the *k* populations have a natural ordering.

- A separate test is performed for each and every continuous and ordinal field specified on the TEST keyword.
- The test specifications given on the `JONCKHEERE_TERPSTRA` keyword apply to all tests performed.

ORDER=ASCENDING| DESCENDING. *Order of the k populations.* `ASCENDING` specifies the alternative hypothesis that the location parameter of the first group is less than or equal to the second, which is less than or equal to the third, and so on. This is the default. `DESCENDING` specifies the alternative hypothesis that the location parameter of the first group is greater than or equal to the second, which is greater than or equal to the third, and so on. For both options, the alternative hypothesis also assumes that the locations are not all equal.

COMPARE=PAIRWISE|STEPWISE|NONE. *Multiple comparisons.* The `COMPARE` keyword controls how and whether multiple comparisons should be performed. `PAIRWISE` produces all pairwise multiple comparisons. `STEPWISE` produces stepwise stepdown comparisons. `NONE` turns off multiple comparisons. By default, all pairwise comparisons are produced.

RELATED Subcommand

The `RELATED` subcommand produces related-samples nonparametric tests. The `TEST` keyword is required, all other keywords are optional. If `RELATED` is specified with none of the optional keywords, the following tests are performed automatically:

- If two categorical fields are specified and they have only two values, McNemar's test is performed.
- If two continuous fields are specified, a Wilcoxon Matched-Pairs Signed-Rank test is performed.
- If more than two categorical fields are specified and they each have only two values, Cochran's test is performed.
- If more than two continuous fields are specified, Friedman's Two-Way Analysis of Variance by Ranks is performed.
- Otherwise, no tests are performed.

When fields of differing measurement level are specified, they are first separated by measurement level and then the appropriate test is applied to each group. For example, if you specify:

```
NPTESTS
  /RELATED TEST(c1 n1 o1 c2 o2 n2 o3 c3 o4).
```

Where *x1*, *x2*, and *x3* are continuous fields, *a1* and *a2* are nominal fields, and *o1*, *o2*, *o3*, and *o4* are ordinal fields, then Friedman's test is applied to the continuous fields, McNemar's test is applied to the nominal fields, and no test is applied to the ordinal fields.

TEST Keyword

The `TEST` keyword lists the fields that you want to test.

- Specify one or more fields. Note that certain tests are not applicable to fields of a particular measurement level; for example, the McNemar test is only performed for categorical fields. NPTESTS automatically determines which tests are applicable to which fields. See the individual keyword descriptions for details.

```
    [MCNEMAR( [SUCCESS={FIRST**        }])]
                       {LIST(valuelist)}
```

The `MCNEMAR` keyword produces a related samples test of whether combinations of values between two dichotomous fields are equally likely.

- A separate test is performed for all of the nominal fields and for all of the ordinal fields specified on the `TEST` keyword.
- The test specifications given on the `MCNEMAR` keyword apply to all tests performed.
- If any test has more than two fields, or any field has more than two values, the test is not performed.

Example

```
NPTESTS /RELATED TEST(n1 n2 n3 o1 o2) MCNEMAR.
```

If *n1*, *n2*, *n3* are nominal fields and *o1* and *o2* are ordinal fields, then no test is performed for the nominal fields and a test is performed for the ordinal fields so long as *o1* and *o2* each have only two values.

SUCCESS= FIRST|LIST(valuelist). The SUCCESS keyword specifies how "success" is defined for categorical fields.

- `FIRST` performs the test using the first value found in the sample to define "success". This option is only applicable to nominal or ordinal fields with only two values; all other categorical fields specified on a `RELATED` subcommand where `FIRST` is used will not be tested. This is the default.
- `LIST` performs the test using the specified list of values to define "success". Specify a list of string or numeric values. The values in the list do not need to be present in the sample.

WILCOXON Keyword

The `WILCOXON` keyword produces a related samples test of whether the distribution of two paired fields in two related samples is the same. The test takes into account the magnitude of the differences between two paired fields.

- A single test is performed for all of the continuous and ordinal fields specified on the TEST keyword.
- If there are more than two continuous or ordinal fields, the test is not performed.

SIGN Keyword

The `SIGN` keyword produces a related samples test of whether the distribution of two paired fields in a two-related-samples test is the same.

- A single test is performed for all of the continuous and ordinal fields specified on the TEST keyword.
- If there are more than two continuous or ordinal fields, the test is not performed.

MARGINAL_HOMOGENEITY Keyword

The `MARGINAL_HOMOGENEITY` keyword produces a related samples test of whether combinations of values between two paired ordinal fields are equally likely. The marginal homogeneity test is typically used in repeated measures situations. This test is an extension of the McNemar test from binary response to multinomial response.

- A single test is performed for all of the ordinal fields specified on the TEST keyword.

- If there are more than two ordinal fields, the test is not performed.

HODGES_LEHMAN Keyword

The `HODGES_LEHMAN` keyword produces a related samples estimate and confidence interval for the median difference between two paired fields.

- A single confidence interval is computed for all of the continuous and ordinal fields specified on the `TEST` keyword.
- If there are more than two continuous or ordinal fields, no confidence intervals are created.

```
    [COCHRAN(
        [SUCCESS={FIRST**        }]
                 {LIST(valuelist)}
        [COMPARE={PAIRWISE**}]
                 {STEPWISE  }
                 {NONE      }
     )]
```

The `COCHRAN` keyword produces a related samples test of whether the distribution of values is the same for *k* related dichotomous fields.

- A separate test is performed for all of the nominal fields and for all of the ordinal fields specified on the `TEST` keyword.
- The test specifications given on the `COCHRAN` keyword apply to all tests performed.
- If any field has more than two values, the test is not performed.

Example

```
NPTESTS /RELATED TEST(n1 n2 n3 o1 o2) COCHRAN.
```

If *n1*, *n2*, *n3* are nominal fields and *o1* and *o2* are ordinal fields, then a test is performed for the nominal fields and a test is performed for the ordinal fields so long as each field has only two values.

SUCCESS= FIRST|LIST(valuelist). The `SUCCESS` keyword specifies how "success" is defined for categorical fields.

- `FIRST` performs the test using the first value found in the sample to define "success". This option is only applicable to nominal or ordinal fields with only two values; all other categorical fields specified on a `RELATED` subcommand where `FIRST` is used will not be tested. This is the default.
- `LIST` performs the test using the specifed list of values to define "success". Specify a list of string or numeric values. The values in the list do not need to be present in the sample.

COMPARE= PAIRWISE| STEPWISE|NONE. *Multiple comparisons.* The `COMPARE` keyword controls how and whether multiple comparisons should be performed. `PAIRWISE` produces all pairwise multiple comparisons. `STEPWISE` produces stepwise stepdown comparisons. `NONE` turns off multiple comparisons. By default, all pairwise comparisons are produced.

FRIEDMAN Keyword

```
    [FRIEDMAN([COMPARE={PAIRWISE**}])]
                       {STEPWISE  }
                       {NONE      }
```

The `FRIEDMAN` subcommand produces a related samples test of whether *k* related samples have been drawn from the same population.

A single test is performed for all of the continuous and ordinal fields specified on the `TEST` keyword.

COMPARE= PAIRWISE| STEPWISE| NONE. *Multiple comparisons.* The `COMPARE` keyword controls how and whether multiple comparisons should be performed. `PAIRWISE` produces all pairwise multiple comparisons. `STEPWISE` produces stepwise stepdown comparisons. `NONE` turns off multiple comparisons. By default, all pairwise comparisons are produced.

KENDALL Keyword

```
     [KENDALL([COMPARE={PAIRWISE**}])]
                       {STEPWISE  }
                       {NONE      }
```

The `KENDALL` keyword produces a related samples test of whether *k* related samples are from the same population. Kendall's *W* is a measure of agreement among judges or raters, where each case is one judge's rating of several items (fields).

A single test is performed for all of the continuous and ordinal fields specified on the `TEST` keyword.

COMPARE= PAIRWISE| STEPWISE| NONE. *Multiple comparisons.* The `COMPARE` keyword controls how and whether multiple comparisons should be performed. `PAIRWISE` produces all pairwise multiple comparisons. `STEPWISE` produces stepwise stepdown comparisons. `NONE` turns off multiple comparisons. By default, all pairwise comparisons are produced.

NPAR TESTS

`NPAR TESTS` is available in Statistics Base Edition.

```
NPAR TESTS [CHISQUARE=varlist[(lo,hi)]/] [/EXPECTED={EQUAL      }]
                                                    {f1,f2,...fn}

 [/K-S({UNIFORM [min,max]   })=varlist]
       {NORMAL [mean,stddev]}
       {POISSON [mean]      } @@
       {EXPONENTIAL [mean]  }

 [/KS_SIM
       [CIN({99**}{value})] [SAMPLES({10000**}{integer})] [NONORMAL] ]

 [/RUNS({MEAN  })=varlist]
        {MEDIAN}
        {MODE  }
        {value }

 [/BINOMIAL[({.5})]=varlist[({value1,value2})]]
             { p}            {value        }

 [/MCNEMAR=varlist [WITH varlist [(PAIRED)]]]

 [/SIGN=varlist [WITH varlist [(PAIRED)]]]

 [/WILCOXON=varlist [WITH varlist [(PAIRED)]]]

    |/MH=varlist [WITH varlist [(PAIRED)]]]††

 [/COCHRAN=varlist]

 [/FRIEDMAN=varlist]

 [/KENDALL=varlist]

 [/M-W=varlist BY var (value1,value2)]

 [/K-S=varlist BY var (value1,value2)]

 [/W-W=varlist BY var (value1,value2)]

 [/MOSES[(n)]=varlist BY var (value1,value2)]

 [/K-W=varlist BY var (value1,value2)]

 [/J-T=varlist BY var (value1, value2)]††

 [/MEDIAN[(value)]=varlist BY var (value1,value2)]

 [/MISSING=[{ANALYSIS**}]  [INCLUDE]]
            {LISTWISE  }

 [/SAMPLE]

 [/STATISTICS=[DESCRIPTIVES]  [QUARTILES] [ALL]]


    [/METHOD={MC [CIN({99.0 })] [SAMPLES({10000})] }]††
                      {value}            {value}
          {EXACT [TIMER({5    })]              }
                         {value}
```

**Default if the subcommand is omitted.

@@ An error message is produced when **`varlist`** is specified for the **`POISSON`** keyword.

††Available only if SPSS Statistics Premium Edition or the Exact Tests option is installed (available only on Windows operating systems).

This command reads the active dataset and causes execution of any pending commands. See the topic Command Order for more information.

Release History

Release 17.0

- Increased limits on number of variables allowed in the analysis.

Release 27

- Added support for the **`KS_SIM`** subcommand.
  - Added support for the **`CIN`** keyword in the **`KS_SIM`** subcommand.
  - Added support for the **`SAMPLES`** keyword in the **`KS_SIM`** subcommand.
  - Added support for the **`NONORMAL`** keyword in the **`KS_SIM`** subcommand.
- Removed support for **`POISSON=varlist`** in the **`K-S`** subcommand.

Example

```
NPAR TESTS K-S(UNIFORM)=V1 /K-S(NORMAL,0,1)=V2.
```

Overview

`NPAR TESTS` is a collection of nonparametric tests. These tests make minimal assumptions about the underlying distribution of the data. [1] In addition to the nonparametric tests that are available in `NPAR TESTS`, the *k*-sample chi-square and Fisher's exact test are available in procedure `CROSSTABS`.

The tests that are available in `NPAR TESTS` can be grouped into three broad categories based on how the data are organized: one-sample tests, related-samples tests, and independent-samples tests. A one-sample test analyzes one variable. A test for related samples compares two or more variables for the same set of cases. An independent-samples test analyzes one variable that is grouped by categories of another variable.

The one-sample tests that are available in procedure `NPAR TESTS` are:

- `BINOMIAL`
- `CHISQUARE`
- `K-S` (Kolmogorov-Smirnov)
- `RUNS`

Tests for two related samples are:

- `MCNEMAR`
- `SIGN`
- `WILCOXON`

Tests for *k* related samples are:

- `COCHRAN`
- `FRIEDMAN`
- `KENDALL`

Tests for two independent samples are:

- `M-W` (Mann-Whitney)
- `K-S` (Kolmogorov-Smirnov)
- `W-W` (Wald-Wolfowitz)

---

[1] Siegel, S., and N. J. Castellan. 1988. *Nonparametric statistics for the behavioral sciences*. New York: McGraw-Hill, Inc..

- `MOSES`

Tests for *k* independent samples are:

- `K-W` (Kruskal-Wallis)
- `MEDIAN`

Options

Statistical Display
In addition to the tests, you can request univariate statistics, quartiles, and counts for all variables that are specified on the command. You can also control the pairing of variables in tests for two related samples.

Random Sampling
`NPAR TESTS` must store cases in memory when computing tests that use ranks. You can use random sampling when there is not enough space to store all cases.

Basic specification

The basic specification is a single test subcommand and a list of variables to be tested. Some tests require additional specifications. `CHISQUARE` has an optional subcommand.

Subcommand order

Subcommands can be used in any order.

Syntax rules

- The `KS_SIM, STATISTICS, SAMPLE,` and `MISSING` subcommands are optional. Each subcommand can be specified only once per `NPAR TESTS` command.
- You can request any or all tests, and you can specify a test subcommand more than once on a single `NPAR TESTS` command.
- If you specify a variable more than once on a test subcommand, only the first variable is used.
- Keyword `ALL` in any variable list refers to all user-defined variables in the active dataset.
- Keyword `WITH` controls pairing of variables in two-related-samples tests.
- Keyword BY introduces the grouping variable in two- and *k*-independent-samples tests.
- Keyword `PAIRED` can be used with keyword `WITH` on the `MCNEMAR`, `SIGN`, and `WILCOXON` subcommands to obtain sequential pairing of variables for two related samples.
- `POISSON = varlist` is not supported in the **`K-S`** subcommand.

Operations

- If a string variable is specified on any subcommand, `NPAR TESTS` will stop executing.
- When `ALL` is used, requests for tests of variables with themselves are ignored and a warning is displayed.

Limitations

- A maximum of 100 subcommands is allowed.
- A maximum of 48,000 variables total per `NPAR TESTS` command is allowed.
- A maximum of 200 values for subcommand `CHISQUARE` is allowed.

BINOMIAL Subcommand

```
NPAR TESTS BINOMIAL [({.5})]=varlist[({value,value})]
                      {p }            {value      }
```

`BINOMIAL` tests whether the observed distribution of a dichotomous variable is the same as what is expected from a specified binomial distribution. By default, each named variable is assumed to have only two values, and the distribution of each named variable is compared to a binomial distribution with *p* (the proportion of cases expected in the first category) equal to 0.5. The default output includes the number of valid cases in each group, the test proportion, and the two-tailed probability of the observed proportion.

Syntax

- The minimum specification is a list of variables to be tested.
- To change the default 0.5 test proportion, specify a value in parentheses immediately after keyword `BINOMIAL`.
- A single value in parentheses following the variable list is used as a cutting point. Cases with values that are equal to or less than the cutting point form the first category; the remaining cases form the second category.
- If two values appear in parentheses after the variable list, cases with values that are equal to the first value form the first category, and cases with values that are equal to the second value form the second category.
- If no values are specified, the variables must be dichotomous. The first value encountered in the dataset defines the first category, and the other value defines the second category.

Operations

- The proportion observed in the first category is compared to the test proportion. The probability of the observed proportion occurring given the test proportion and a binomial distribution is then computed. A test statistic is calculated for each variable specified.
- If the test proportion is the default (0.5), a two-tailed probability is displayed. For any other test proportion, a one-tailed probability is displayed. The direction of the one-tailed test depends on the observed proportion in the first category. If the observed proportion is more than the test proportion, the significance of observing that many or more in the first category is reported. If the observed proportion is less than or equal to the test proportion, the significance of observing that many or fewer in the first category is reported. In other words, the test is always done in the observed direction.

Example

```
NPAR TESTS BINOMIAL(.667)=V1(0,1).
```

- `NPAR TESTS` displays the Binomial Test table, showing the number of cases, observed proportion, test proportion (0.667), and the one-tailed significance for each category.
- If more than 0.667 of the cases have value 0 for *V1*, `BINOMIAL` gives the probability of observing that many or more values of 0 in a binomial distribution with probability 0.667. If fewer than 0.667 of the cases are 0, the test will be of observing that many or fewer values.

CHISQUARE Subcommand

```
NPAR TESTS CHISQUARE=varlist [(lo,hi)] [/EXPECTED={EQUAL**     }]
                                                   {f1,f2,..., fn}
```

The `CHISQUARE` (alias `CHI-SQUARE`) one-sample test computes a chi-square statistic based on the differences between the observed and expected frequencies of categories of a variable. By default, equal frequencies are expected in each category. The output includes the frequency distribution, expected frequencies, residuals, chi-square, degrees of freedom, and probability.

Syntax

- The minimum specification is a list of variables to be tested. Optionally, you can specify a value range in parentheses following the variable list. You can also specify expected proportions with the `EXPECTED` subcommand.
- If you use the `EXPECTED` subcommand to specify unequal expected frequencies, you must specify a value greater than 0 for each observed category of the variable. The expected frequencies are specified

in ascending order of category value. You can use the notation *n*f* to indicate that frequency *f* is expected for *n* consecutive categories.

- Specifying keyword `EQUAL` on the `EXPECTED` subcommand has the same effect as omitting the `EXPECTED` subcommand.
- `EXPECTED` applies to all variables that are specified on the `CHISQUARE` subcommand. Use multiple `CHISQUARE` and `EXPECTED` subcommands to specify different expected proportions for variables.

Operations

- If no range is specified for the variables that are to be tested, a separate Chi-Square Frequency table is produced for each variable. Each distinct value defines a category.
- If a range is specified, integer-valued categories are established for each value within the range. Non-integer values are truncated before classification. Cases with values that are outside the specified range are excluded. One combined Chi-Square Frequency table is produced for all specified variables.
- Expected values are interpreted as proportions, not absolute values. Values are summed, and each value is divided by the total to calculate the proportion of cases expected in the corresponding category.
- A test statistic is calculated for each specified variable.

Example

```
NPAR TESTS CHISQUARE=V1 (1,5) /EXPECTED= 12, 3*16, 18.
```

- This example requests the chi-square test for values 1 through 5 of variable *V1*.
- The observed frequencies for variable *V1* are compared with the hypothetical distribution of 12/78 occurrences of value 1; 16/78 occurrences each of values 2, 3, and 4; and 18/78 occurrences of value 5.

COCHRAN Subcommand

```
NPAR TESTS COCHRAN=varlist
```

`COCHRAN` calculates Cochran's *Q,* which tests whether the distribution of values is the same for *k* related dichotomous variables. The output shows the frequency distribution for each variable in the Cochran Frequencies table and the number of cases, Cochran's *Q*, degrees of freedom, and probability in the Test Statistics table.

Syntax

- The minimum specification is a list of two variables.
- The variables must be dichotomous and must be coded with the same two values.

Operations

- A *k* × 2 contingency table (variables by categories) is constructed for dichotomous variables, and the proportions for each variable are computed. A single test is calculated, comparing all variables.
- Cochran's *Q* statistic has approximately a chi-square distribution.

Example

```
NPAR TESTS COCHRAN=RV1 TO RV3.
```

- This example tests whether the distribution of values 0 and 1 for *RV1*, *RV2*, and *RV3* is the same.

FRIEDMAN Subcommand

```
NPAR TESTS FRIEDMAN=varlist
```

`FRIEDMAN` tests whether *k* related samples have been drawn from the same population. The output shows the mean rank for each variable in the Friedman Ranks table and the number of valid cases, chi-square, degrees of freedom, and probability in the Test Statistics table.

Syntax

- The minimum specification is a list of two variables.
- Variables should be at least at the ordinal level of measurement.

Operations

- The values of *k* variables are ranked from 1 to *k* for each case, and the mean rank is calculated for each variable over all cases.
- The test statistic has approximately a chi-square distribution. A single test statistic is calculated, comparing all variables.

Example

```
NPAR TESTS FRIEDMAN=V1 V2 V3
  /STATISTICS=DESCRIPTIVES.
```

- This example tests variables *V1, V2*, and *V3*, and the example requests univariate statistics for all three variables.

J-T Subcommand

```
NPAR TESTS /J-T=varlist BY variable(value1,value2)
```

`J-T` (alias `JONCKHEERE-TERPSTRA`) performs the Jonckheere-Terpstra test, which tests whether *k* independent samples that are defined by a grouping variable are from the same population. This test is particularly powerful when the *k* populations have a natural ordering. The output shows the number of levels in the grouping variable; the total number of cases; observed, standardized, mean, and standard deviation of the test statistic; the two-tailed asymptotic significance; and, if a `/METHOD` subcommand is specified, one-tailed and two-tailed exact or Monte Carlo probabilities. This subcommand is available only if SPSS Statistics Premium Edition or the Exact Tests option is installed.

Syntax

- The minimum specification is a test variable, the keyword BY, a grouping variable, and a pair of values in parentheses.
- Every value in the range defined by the pair of values for the grouping variable forms a group.
- If the `/ METHOD` subcommand is specified, and the number of populations, *k*, is greater than 5, the *p* value is estimated by using the Monte Carlo sampling method. The exact *p* value is not available when *k* exceeds 5.

Operations

- Cases from the *k* groups are ranked in a single series, and the rank sum for each group is computed. A test statistic is calculated for each variable that is specified before BY.
- The Jonckheere-Terpstra statistic has approximately a normal distribution.
- Cases with values other than values in the range that is specified for the grouping variable are excluded.
- The direction of a one-tailed inference is indicated by the sign of the standardized test statistic.

Example

```
NPAR TESTS /J-T=V1 BY V2(0,4)
  /METHOD=EXACT.
```

- This example performs the Jonckheere-Terpstra test for groups that are defined by values 0 through 4 of *V2*. The exact *p* values are calculated.

K-S Subcommand (One-Sample)

```
NPAR TESTS K-S({NORMAL [mean,stddev]})=varlist
               {POISSON [mean]      }
               {UNIFORM [min,max]   }
               {EXPONENTIAL [mean]  }
```

The `K-S` (alias `KOLMOGOROV-SMIRNOV`) one-sample test compares the cumulative distribution function for a variable with a uniform, normal, Poisson, or exponential distribution, and the test tests whether the distributions are homogeneous. The parameters of the test distribution can be specified; the defaults are the observed parameters. The output shows the number of valid cases, parameters of the test distribution, most-extreme absolute, positive, and negative differences, Kolmogorov-Smirnov *Z*, and two-tailed probability for each variable.

Syntax

The minimum specification is a distribution keyword and a list of variables. The distribution keywords are `NORMAL`, `POISSON`, `EXPONENTIAL`, and `UNIFORM`.

- The distribution keyword and its optional parameters must be enclosed within parentheses.
- The distribution keyword must be separated from its parameters by blanks or commas.

NORMAL [mean, stdev]. *Normal distribution.* The default parameters are the observed mean and standard deviation.

POISSON [mean]. *Poisson distribution.* The default parameter is the observed mean.

Note: `POISSON = varlist` is not supported in the **`K-S`** subcommand.

UNIFORM [min,max]. *Uniform distribution.* The default parameters are the observed minimum and maximum values.

EXPONENTIAL [mean]. *Exponential distribution.* The default parameter is the observed mean.

Operations

- The Kolmogorov-Smirnov *Z* is computed from the largest difference in absolute value between the observed and test distribution functions.
- The K-S probability levels assume that the test distribution is specified entirely in advance. The distribution of the test statistic and resulting probabilities are different when the parameters of the test distribution are estimated from the sample. The K-S test for normality uses the Lilliefors adjustment if you do not enter a population mean and standard deviation. The power of the test to detect departures from the hypothesized distribution may be seriously diminished. For testing against a normal distribution with estimated parameters, consider the adjusted K-S Lilliefors test that is available in the `EXAMINE` procedure.
- For a mean of 100,000 or larger, a normal approximation to the Poisson distribution is used.
- A test statistic is calculated for each specified variable.

Example

```
NPAR TESTS K-S(UNIFORM)=V1 /K-S(NORMAL,0,1)=V2.
```

- The first `K-S` subcommand compares the distribution of *V1* with a uniform distribution that has the same range as *V1*.
- The second `K-S` subcommand compares the distribution of *V2* with a normal distribution that has a mean of 0 and a standard deviation of 1.

K-S Subcommand (Two-Sample)

```
NPAR TESTS K-S=varlist BY variable(value1,value2)
```

`K-S` (alias `KOLMOGOROV-SMIRNOV`) tests whether the distribution of a variable is the same in two independent samples that are defined by a grouping variable. The test is sensitive to any difference in median, dispersion, skewness, and so forth, between the two distributions. The output shows the valid number of cases in each group in the Frequency table. The output also shows the largest absolute, positive, and negative differences between the two groups, the Kolmogorov-Smirnov *Z*, and the two-tailed probability for each variable in the Test Statistics table.

Syntax

- The minimum specification is a test variable, the keyword BY, a grouping variable, and a pair of values in parentheses.
- The test variable should be at least at the ordinal level of measurement.
- Cases with the first value form one group, and cases with the second value form the other group. The order in which values are specified determines which difference is the largest positive and which difference is the largest negative.

Operations

- The observed cumulative distributions are computed for both groups, as are the maximum positive, negative, and absolute differences. A test statistic is calculated for each variable that is named before BY.
- Cases with values other than values that are specified for the grouping variable are excluded.

Example

```
NPAR TESTS K-S=V1 V2 BY V3(0,1).
```

- This example specifies two tests. The first test compares the distribution of *V1* for cases with value 0 for *V3* with the distribution of *V1* for cases with value 1 for *V3*.
- A parallel test is calculated for *V2*.

KS_SIM Subcommand (One-Sample)

```
NPAR TESTS
   [K-S({NORMAL=varlist
          {POISSON}
          {UNIFORM}
          {EXPONENTIAL}
   [/KS_SIM ({CIN [value]} {SAMPLES [integer]} {NONORMAL}) ]
```

The optional `KS_SIM` (alias `KOLMOGOROV-SMIRNOV simulation`) one-sample subcommand takes effect when a continuous distribution, **`Normal`**, **`Uniform`**, or **`Exponential`**, is specified on the **`K-S`** one-sample subcommand (without parameters being specified). The `KS_SIM` subcommand controls the parameters for the Monte Carlo simulation for Normal, Uniform, and Exponential distributions.

Syntax

CIN [value]. *Confidence interval.* Resets the estimated confidence interval level used by the Kolmogorov-Smirnov test (using the Monte Carlo simulations).

SAMPLES [integer]. *Monte Carlo sampling.* resets the number of replicates used by the Lilliefors test for Monte Carlo sampling.

NONORMAL. *Do not include Monte Carlo sampling.* When specified, results will not include the Monte Carlo sampling for Normal distribution.

Example

```
NPAR TESTS
  /K-S(NORMAL)=agecat gender
  /K-S(UNIFORM)=agecat gender
  /K-S(POISSON 1)=agecat gender
  /K-S(EXPONENTIAL)=agecat gender
  /KS_SIM CIN(99) SAMPLES(10000).
```

K-W Subcommand

```
NPAR TESTS K-W=varlist BY variable(value1,value2)
```

`K-W` (alias `KRUSKAL-WALLIS`) tests whether *k* independent samples that are defined by a grouping variable are from the same population. The output shows the number of valid cases and the mean rank of the variable in each group in the Ranks table. the output also shows the chi-square, degrees of freedom, and probability in the Test Statistics table.

Syntax

- The minimum specification is a test variable, the keyword BY, a grouping variable, and a pair of values in parentheses.
- Every value in the range defined by the pair of values for the grouping variable forms a group.

Operations

- Cases from the *k* groups are ranked in a single series, and the rank sum for each group is computed. A test statistic is calculated for each variable that is specified before BY.
- Kruskal-Wallis *H* has approximately a chi-square distribution.
- Cases with values other than values in the range that is specified for the grouping variable are excluded.

Example

```
NPAR TESTS K-W=V1 BY V2(0,4).
```

- This example tests *V1* for groups that are defined by values 0 through 4 of *V2*.

KENDALL Subcommand

```
NPAR TESTS KENDALL=varlist
```

`KENDALL` tests whether *k* related samples are from the same population. *W* is a measure of agreement among judges or raters, where each case is one judge's rating of several items (variables). The output includes the mean rank for each variable in the Ranks table and the valid number of cases, Kendall's *W*, chi-square, degrees of freedom, and probability in the Test Statistics table.

Syntax

The minimum specification is a list of two variables.

Operations

- The values of the *k* variables are ranked from 1 to *k* for each case, and the mean rank is calculated for each variable over all cases. Kendall's *W* and a corresponding chi-square statistic are calculated, correcting for ties. In addition, a single test statistic is calculated for all variables.
- *W* ranges between 0 (no agreement) and 1 (complete agreement).

Example

```
DATA LIST /V1 TO V5 1-10.
BEGIN DATA
2 5 4 5 1
3 3 4 5 3
3 4 4 6 2
2 4 3 6 2
END DATA.
NPAR TESTS KENDALL=ALL.
```

- This example tests four judges (cases) on five items (variables *V1* through *V5*).

M-W Subcommand

```
NPAR TESTS M-W=varlist BY variable(value1,value2)
```

`M-W` (alias `MANN-WHITNEY`) tests whether two independent samples that are defined by a grouping variable are from the same population. The test statistic uses the rank of each case to test whether the groups are drawn from the same population. The output shows the number of valid cases of each group; the mean rank of the variable within each group and the sum of ranks in the Ranks table and the Mann-Whitney *U*; Wilcoxon *W* (the rank sum of the smaller group); *Z* statistic; and probability in the Test Statistics table.

Syntax

- The minimum specification is a test variable, the keyword BY, a grouping variable, and a pair of values in parentheses.

- Cases with the first value form one group and cases with the second value form the other group. The order in which the values are specified is unimportant.

Operations

- Cases are ranked in order of increasing size, and test statistic *U* (the number of times that a score from group 1 precedes a score from group 2) is computed.
- An exact significance level is computed if there are 40 or fewer cases. For more than 40 cases, *U* is transformed into a normally distributed *Z* statistic, and a normal approximation *p* value is computed.
- A test statistic is calculated for each variable that is named before `BY`.
- Cases with values other than values that are specified for the grouping variable are excluded.

Example

```
NPAR TESTS M-W=V1 BY V2(1,2).
```

- This example tests *V1* based on the two groups that are defined by values 1 and 2 of *V2*.

MCNEMAR Subcommand

```
NPAR TESTS MCNEMAR=varlist [WITH varlist [(PAIRED)]]
```

`MCNEMAR` tests whether combinations of values between two dichotomous variables are equally likely. The output includes a Crosstabulation table for each pair and a Test Statistics table for all pairs, showing the number of valid cases, chi-square, and probability for each pair.

Syntax

- The minimum specification is a list of two variables. Variables must be dichotomous and must have the same two values.
- If keyword `WITH` is not specified, each variable is paired with every other variable in the list.
- If `WITH` is specified, each variable before `WITH` is paired with each variable after `WITH`. If `PAIRED` is also specified, the first variable before `WITH` is paired with the first variable after `WITH`, the second variable before `WITH` is paired with the second variable after `WITH`, and so on. `PAIRED` cannot be specified without `WITH`.
- With `PAIRED`, the number of variables that are specified before and after `WITH` must be the same. `PAIRED` must be specified in parentheses after the second variable list.

Operations

- For the purposes of computing the test statistics, only combinations for which the values for the two variables are different are considered.
- If fewer than 25 cases change values from the first variable to the second variable, the binomial distribution is used to compute the probability.

Example

```
NPAR TESTS MCNEMAR=V1 V2 V3.
```

- This example performs the `MCNEMAR` test on variable pairs *V1* and *V2*, *V1* and *V3*, and *V2* and *V3*.

MEDIAN Subcommand

```
NPAR TESTS MEDIAN [(value)]=varlist BY variable(value1,value2)
```

`MEDIAN` determines whether *k* independent samples are drawn from populations with the same median. The independent samples are defined by a grouping variable. For each variable, the output shows a table of the number of cases that are greater than and less than or equal to the median in each category in the Frequency table. The output also shows the number of valid cases, the median, chi-square, degrees of freedom, and probability in the Test Statistics table.

Syntax

- The minimum specification is a single test variable, the keyword BY, a grouping variable, and two values in parentheses.
- If the first grouping value is less than the second value, every value in the range that is defined by the pair of values forms a group, and a *k*-sample test is performed.
- If the first value is greater than the second value, two groups are formed by using the two values, and a two-sample test is performed.
- By default, the median is calculated from all cases that are included in the test. To override the default, specify a median value in parentheses following the `MEDIAN` subcommand keyword.

Operations

- A 2 × *k* contingency table is constructed with counts of the number of cases that are greater than the median and less than or equal to the median for the *k* groups.
- Test statistics are calculated for each variable that is specified before BY.
- For more than 30 cases, a chi-square statistic is computed. For 30 or fewer cases, Fisher's exact procedure (two-tailed) is used instead of chi-square.
- For a two-sample test, cases with values other than the two specified values are excluded.

Example

```
NPAR TESTS MEDIAN(8.4)=V1 BY V2(1,2) /MEDIAN=V1 BY V2(1,2)
  /MEDIAN=V1 BY V3(1,4) /MEDIAN=V1 BY V3(4,1).
```

- The first two `MEDIAN` subcommands test variable *V1* grouped by values 1 and 2 of variable *V2*. The first test specifies a median of 8.4, and the second test uses the observed median.
- The third `MEDIAN` subcommand requests a four-samples test, dividing the sample into four groups based on values 1, 2, 3, and 4 of variable *V3*.
- The last `MEDIAN` subcommand requests a two-samples test, grouping cases based on values 1 and 4 of *V3* and ignoring all other cases.

MH Subcommand

```
NPAR TESTS /MH=varlist [WITH varlist [(PAIRED)]]
```

`MH` performs the marginal homogeneity test, which tests whether combinations of values between two paired ordinal variables are equally likely. The marginal homogeneity test is typically used in repeated measures situations. This test is an extension of the McNemar test from binary response to multinomial response. The output shows the number of distinct values for all test variables; the number of valid off-diagonal cell counts; mean; standard deviation; observed and standardized values of the test statistics; the asymptotic two-tailed probability for each pair of variables; and, if a `/METHOD` subcommand is specified, one-tailed and two-tailed exact or Monte Carlo probabilities.

This subcommand is available only if SPSS Statistics Premium Edition or the Exact Tests option is installed (available only on Windows operating systems).

Syntax

- The minimum specification is a list of two variables. Variables must be polychotomous and must have more than two values. If the variables contain only two values, the McNemar test is performed.
- If keyword `WITH` is not specified, each variable is paired with every other variable in the list.
- If `WITH` is specified, each variable before `WITH` is paired with each variable after `WITH`. If `PAIRED` is also specified, the first variable before `WITH` is paired with the first variable after `WITH`, the second variable before `WITH` is paired with the second variable after `WITH`, and so on. `PAIRED` cannot be specified without `WITH`.
- With `PAIRED`, the number of variables that are specified before and after `WITH` must be the same. `PAIRED` must be specified in parentheses after the second variable list.

Operations

- The data consist of paired, dependent responses from two populations. The marginal homogeneity test tests the equality of two multinomial *c* × 1 tables, and the data can be arranged in the form of a square *c* × *c* contingency table. A 2 × *c* table is constructed for each off-diagonal cell count. The marginal homogeneity test statistic is computed for cases with different values for the two variables. Only combinations for which the values for the two variables are different are considered. The first row of each 2 × *c* table specifies the category that was chosen by population 1, and the second row specifies the category that was chosen by population 2. The test statistic is calculated by summing the first row scores across all 2 x *c* tables.

Example

```
NPAR TESTS /MH=V1 V2 V3
  /METHOD=MC.
```

- This example performs the marginal homogeneity test on variable pairs *V1* and *V2*, *V1* and *V3*, and *V2* and *V3*. The exact *p* values are estimated by using the Monte Carlo sampling method.

MOSES Subcommand

```
NPAR TESTS MOSES[(n)]=varlist BY variable(value1,value2)
```

The `MOSES` test of extreme reactions tests whether the range of an ordinal variable is the same in a control group and a comparison group. The control and comparison groups are defined by a grouping variable. The output includes a Frequency table, showing, for each variable before *BY*, the total number of cases and the number of cases in each group. The output also includes a Test Statistics table, showing the number of removed outliers, span of the control group before and after outliers are removed, and one-tailed probability of the span with and without outliers.

Syntax

- The minimum specification is a test variable, the keyword BY, a grouping variable, and two values in parentheses.
- The test variable must be at least at the ordinal level of measurement.
- The first value of the grouping variable defines the control group, and the second value defines the comparison group.
- By default, 5% of the cases are trimmed from each end of the range of the control group to remove outliers. You can override the default by specifying a value in parentheses following the `MOSES` subcommand keyword. This value represents an actual number of cases, not a percentage.

Operations

- Values from the groups are arranged in a single ascending sequence. The span of the control group is computed as the number of cases in the sequence containing the lowest and highest control values.
- No adjustments are made for tied cases.
- Cases with values other than values that are specified for the grouping variable are excluded.
- Test statistics are calculated for each variable that is named before BY.

Example

```
NPAR TESTS MOSES=V1 BY V3(0,1) /MOSES=V1 BY V3(1,0).
```

- The first `MOSES` subcommand tests *V1* by using value 0 of *V3* to define the control group and value 1 for the comparison group. The second `MOSES` subcommand reverses the comparison and control groups.

RUNS Subcommand

```
NPAR TESTS RUNS({MEAN  })=varlist
                {MEDIAN}
                {MODE  }
                {value }
```

`RUNS` tests whether the sequence of values of a dichotomized variable is random. The output includes a Run Test table, showing the test value (cut point that is used to dichotomize the variable tested), number of runs, number of cases that are below the cut point, number of cases that are greater than or equal to the cut point, and test statistic *Z* with its two-tailed probability for each variable.

Syntax

- The minimum specification is a cut point in parentheses followed by a test variable.
- The cut point can be specified by an exact value or one of the keywords `MEAN`, `MEDIAN`, or `MODE`.

Operations

- All tested variables are treated as dichotomous: cases with values that are less than the cut point form one category, and cases with values that are greater than or equal to the cut point form the other category.
- Test statistics are calculated for each specified variable.

Example

```
NPAR TESTS RUNS(MEDIAN)=V2 /RUNS(24.5)=V2 /RUNS(1)=V3.
```

- This example performs three runs tests. The first test tests variable *V2* by using the median as the cut point. The second test also tests *V2* by using 24.5 as the cut point. The third test tests variable *V3*, with value 1 specified as the cut point.

SIGN Subcommand

```
NPAR TESTS SIGN=varlist [WITH varlist [(PAIRED)] ]
```

`SIGN` tests whether the distribution of two paired variables in a two-related-samples test is the same. The output includes a Frequency table, showing, for each pair, the number of positive differences, number of negative differences, number of ties, and the total number. The output also includes a Test Statistics table, showing the *Z* statistic and two-tailed probability.

Syntax

- The minimum specification is a list of two variables.
- Variables should be at least at the ordinal level of measurement.
- If keyword `WITH` is not specified, each variable in the list is paired with every other variable in the list.
- If keyword `WITH` is specified, each variable before `WITH` is paired with each variable after `WITH`. If `PAIRED` is also specified, the first variable before `WITH` is paired with the first variable after `WITH`, the second variable before `WITH` is paired with the second variable after `WITH`, and so on. `PAIRED` cannot be specified without `WITH`.
- With `PAIRED`, the number of variables that are specified before and after `WITH` must be the same. `PAIRED` must be specified in parentheses after the second variable list.

Operations

- The positive and negative differences between the pair of variables are counted. Ties are ignored.
- The probability is taken from the binomial distribution if 25 or fewer differences are observed. Otherwise, the probability comes from the *Z* distribution.
- Under the null hypothesis for large sample sizes, *Z* is approximately normally distributed with a mean of 0 and a variance of 1.

Example

```
NPAR TESTS SIGN=N1,M1 WITH N2,M2 (PAIRED).
```

- *N1* is tested with *N2*, and *M1* is tested with *M2*.

W-W Subcommand

```
NPAR TESTS W-W=varlist BY variable(value1,value2)
```

`W-W` (alias `WALD-WOLFOWITZ`) tests whether the distribution of a variable is the same in two independent samples. A runs test is performed with group membership as the criterion. The output includes a Frequency table, showing the total number of valid cases for each variable that is specified before BY and the number of valid cases in each group. The output also includes a Test Statistics table, showing the number of runs, *Z*, and one-tailed probability of *Z*. If ties are present, the minimum and maximum number of possible runs, their *Z* statistics, and one-tailed probabilities are displayed.

Syntax

- The minimum specification is a single test variable, the keyword BY, a grouping variable, and two values in parentheses.
- Cases with the first value form one group, and cases with the second value form the other group. The order in which values are specified is unimportant.

Operations

- Cases are combined from both groups and ranked from lowest to highest, and a runs test is performed, using group membership as the criterion. For ties involving cases from both groups, both the minimum and maximum number of possible runs are calculated. Test statistics are calculated for each variable that is specified before BY.
- For a sample size of 30 or less, the exact one-tailed probability is calculated. For a sample size that is greater than 30, the normal approximation is used.
- Cases with values other than values that are specified for the grouping variable are excluded.

Example

```
NPAR TESTS W-W=V1 BY V3(0,1).
```

- This example ranks cases from lowest to highest based on their values for *V1*, and a runs test is performed. Cases with value 0 for *V3* form one group, and cases with value 1 form the other group.

WILCOXON Subcommand

```
NPAR TESTS WILCOXON=varlist [WITH varlist [(PAIRED)] ]
```

`WILCOXON` tests whether the distribution of two paired variables in two related samples is the same. This test takes into account the magnitude of the differences between two paired variables. The output includes a Ranks table, showing, for each pair, the number of valid cases, positive and negative differences, their respective mean and sum of ranks, and the number of ties. The output also includes a Test Statistics table, showing *Z* and probability of *Z*.

Syntax

- The minimum specification is a list of two variables.
- If keyword `WITH` is not specified, each variable is paired with every other variable in the list.
- If keyword `WITH` is specified, each variable before `WITH` is paired with each variable after `WITH`. If `PAIRED` is also specified, the first variable before `WITH` is paired with the first variable after `WITH`, the second variable before `WITH` is paired with the second variable after `WITH`, and so on. `PAIRED` cannot be specified without `WITH`.
- With `PAIRED`, the number of variables that are specified before and after `WITH` must be the same. `PAIRED` must be specified in parentheses after the second variable list.

Operations

- The differences between the pair of variables are counted, the absolute differences are ranked, the positive and negative ranks are summed, and the test statistic *Z* is computed from the positive and negative rank sums.

- Under the null hypothesis for large sample sizes, *Z* is approximately normally distributed with a mean of 0 and a variance of 1.

Example

```
NPAR TESTS WILCOXON=A B WITH C D (PAIRED).
```

- This example pairs *A* with *C* and *B* with *D*. If `PAIRED` were not specified, the example would also pair *A* with *D* and *B* with *C*.

STATISTICS Subcommand

`STATISTICS` requests summary statistics for variables that are named on the `NPAR TESTS` command. Summary statistics are displayed in the Descriptive Statistics table before all test output.

- If `STATISTICS` is specified without keywords, univariate statistics (keyword `DESCRIPTIVES`) are displayed.

DESCRIPTIVES. *Univariate statistics.* The displayed statistics include the mean, maximum, minimum, standard deviation, and number of valid cases for each variable named on the command.

QUARTILES. *Quartiles and number of cases.* The 25th, 50th, and 75th percentiles are displayed for each variable that is named on the command.

ALL. *All statistics available on* `NPAR TESTS`.

MISSING Subcommand

`MISSING` controls the treatment of cases with missing values.

- `ANALYSIS` and `LISTWISE` are alternatives. However, each of those commands can be specified with `INCLUDE`.

ANALYSIS. *Exclude cases with missing values on a test-by-test basis.* Cases with missing values for a variable that is used for a specific test are omitted from that test. On subcommands that specify several tests, each test is evaluated separately. This setting is the default.

LISTWISE. *Exclude cases with missing values listwise.* Cases with missing values for any variable that is named on any subcommand are excluded from all analyses.

INCLUDE. *Include user-missing values.* User-missing values are treated as valid values.

SAMPLE Subcommand

`NPAR TESTS` must store cases in memory. `SAMPLE` allows you to select a random sample of cases when there is not enough space on your computer to store all cases. `SAMPLE` has no additional specifications.

- Because sampling would invalidate a runs test, this option is ignored when the `RUNS` subcommand is used.

METHOD Subcommand

`METHOD` displays additional results for each requested statistic. If no `METHOD` subcommand is specified, the standard asymptotic results are displayed. If fractional weights have been specified, results for all methods will be calculated on the weight rounded to the nearest integer. This subcommand is available only if you have the Exact Tests add-on option installed, which is only available on Windows operating systems.

MC. Displays an unbiased point estimate and confidence interval, based on the Monte Carlo sampling method, for all statistics. Asymptotic results are also displayed. When exact results can be calculated, they will be provided instead of the Monte Carlo results. See Exact Tests for situations under which exact results are provided instead of Monte Carlo results.

CIN(n). Controls the confidence level for the Monte Carlo estimate. CIN is available only when /METHOD=MC is specified. CIN has a default value of 99.0. You can specify a confidence interval between 0.01 and 99.9, inclusive.

SAMPLES. Specifies the number of tables that were sampled from the reference set when calculating the Monte Carlo estimate of the exact *p* value. Larger sample sizes lead to narrower confidence limits but also take longer to calculate. You can specify any integer between 1 and 1,000,000,000 as the sample size. SAMPLES has a default value of 10,000.

EXACT. Computes the exact significance level for all statistics, in addition to the asymptotic results. If both the EXACT and MC keywords are specified, only exact results are provided. Calculating the exact *p* value can be memory-intensive. If you have specified /METHOD=EXACT and find that you have insufficient memory to calculate results, close any other applications that are currently running. You can also enlarge the size of your swap file (see your Windows manual for more information). If you still cannot obtain exact results, specify /METHOD=MC to obtain the Monte Carlo estimate of the exact *p* value. An optional TIMER keyword is available if you choose /METHOD=EXACT.

TIMER(n). Specifies the maximum number of minutes during which the exact analysis for each statistic can run. If the time limit is reached, the test is terminated, no exact results are provided, and the program begins to calculate the next test in the analysis. TIMER is available only when /METHOD=EXACT is specified. You can specify any integer value for TIMER. Specifying a value of 0 for TIMER turns the timer off completely. TIMER has a default value of 5 minutes. If a test exceeds a time limit of 30 minutes, it is recommended that you use the Monte Carlo method, rather than the exact method.

References

Siegel, S., and N. J. Castellan. 1988. *Nonparametric statistics for the behavioral sciences*. New York: McGraw-Hill, Inc..

ONEWAY

ONEWAY is available in the Statistics Base Edition.

```
ONEWAY  varlist BY varname

 [/POLYNOMIAL=n]  [/CONTRAST=coefficient list] [/CONTRAST=... ]

 [/POSTHOC=([SNK] [TUKEY] [BTUKEY] [DUNCAN] [SCHEFFE] [DUNNETT[refcat)]
            [DUNNETTL(refcat)] [DUNNETTR(refcat)] [BONFERRONI] [LSD]
            [SIDAK] [GT2] [GABRIEL] [FREGW] [QREGW] [T2] [T3] [GH] [C]
            [WALLER({100** })]) [ALPHA({0.05**})]
                    {Kratio}           {α     }

 [/RANGES={LSD         }([{0.05**}])] [/RANGES=...]
          {DUNCAN      }  {α     }
          {SNK         }
          {TUKEYB      }
          {TUKEY       }
          {MODLSD      }
          {SCHEFFE     }

 [/STATISTICS=[NONE**] [DESCRIPTIVES]  [EFFECTS]  [HOMOGENEITY]  [ALL]  ]
              [WELCH] [BROWNFORSYTHE]

 [/PLOT MEANS ]

 [/MISSING=[{ANALYSIS**}]  [{EXCLUDE**}] ]
            {LISTWISE  }    {INCLUDE  }

 [/MATRIX =[IN({*                  })] [OUT({*                  })]  [NONE] ]
              {'savfile'|'dataset'}         {'savfile'|'dataset'}

 [/TEMPLATE='filename']

 [/CRITERIA=CILEVEL({0.95**}{value})  ]

 [/ES=[OVERALL] [CONTRAST({POOLED**}{GROUP})]  ],
```

**Default if the subcommand is omitted.

This command reads the active dataset and causes execution of any pending commands. See the topic Command Order for more information.

Release History

Release 19.0

- `TEMPLATE` subcommand introduced.

Release 27.0

- `CRITERIA` subcommand introduced.
- `ES` subcommand introduced.

Example

```
ONEWAY V1 BY V2.
```

Overview

`ONEWAY` produces a one-way analysis of variance for an interval-level dependent variable by one numeric independent variable that defines the groups for the analysis. Other procedures that perform an analysis of variance are `SUMMARIZE, UNIANOVA,` and `GLM` (`GLM` is available in SPSS Statistics Standard Edition or the Advanced Statistics Option). Some tests not included in the other procedures are available as options in `ONEWAY`.

Options

Trend and contrasts
You can partition the between-groups sums of squares into linear, quadratic, cubic, and higher-order trend components using the `POLYNOMIAL` subcommand. You can specify up to 10 contrasts to be tested with the *t* statistic on the `CONTRAST` subcommand.

Post hoc tests
You can specify 20 different post hoc tests for comparisons of all possible pairs of group means or multiple comparisons using the `POSTHOC` subcommand.

Statistical display
In addition to the default display, you can obtain means, standard deviations, and other descriptive statistics for each group using the `STATISTICS` subcommand. Fixed- and random-effects statistics as well as Levene tests for homogeneity of variance are also available.

Matrix input and output
You can write means, standard deviations, and category frequencies to a matrix data file that can be used in subsequent `ONEWAY` procedures using the `MATRIX` subcommand. You can also read matrix materials consisting of means, category frequencies, pooled variance, and degrees of freedom for the pooled variance.

Templates
You can specify a template, using the **`TEMPLATE`** subcommand, to override the default chart attribute settings on your system.

Criteria
Specifies the options for the procedure. The **`CILEVEL`** keyword controls the significance level to estimate the confidence intervals.

Effect size estimation
Controls the effect size estimation. The **`OVERALL`** keyword controls the calculation of the effect size for the overall test; the **`CONTRAST`** keyword controls the calculation of the effect size for the contrast test.

Basic Specification

The basic specification is a dependent variable, keyword `BY`, and an independent variable. `ONEWAY` produces an ANOVA table displaying the between- and within-groups sums of squares, mean squares, degrees of freedom, the *F* ratio, and the probability of *F* for each dependent variable by the independent variable.

Subcommand Order

- The variable list must be specified first.
- The remaining subcommands can be specified in any order.

Operations

- All values of the independent variable are used. Each different value creates one category.
- If a string variable is specified as an independent or dependent variable, `ONEWAY` is not executed.

Limitations

- Maximum 100 dependent variables and 1 independent variable.
- An unlimited number of categories for the independent variable. However, post hoc tests are not performed if the number of nonempty categories exceeds 50. Contrast tests are not performed if the total of empty and nonempty categories exceeds 50.
- Maximum 1 `POLYNOMIAL` subcommand.
- Maximum 1 `POSTHOC` subcommand.
- Maximum 10 `CONTRAST` subcommands.

Analysis List

The analysis list consists of a list of dependent variables, keyword BY, and an independent (grouping) variable.

- Only one analysis list is allowed, and it must be specified before any of the optional subcommands.
- All variables named must be numeric.

POLYNOMIAL Subcommand

`POLYNOMIAL` partitions the between-groups sums of squares into linear, quadratic, cubic, or higher-order trend components. The display is an expanded analysis-of-variance table that provides the degrees of freedom, sums of squares, mean square, *F*, and probability of *F* for each partition.

- The value specified on `POLYNOMIAL` indicates the highest-degree polynomial to be used.
- The polynomial value must be a positive integer less than or equal to 5 and less than the number of groups. If the polynomial specified is greater than the number of groups, the highest-degree polynomial possible is assumed.
- Only one `POLYNOMIAL` subcommand can be specified per `ONEWAY` command. If more than one is used, only the last one specified is in effect.
- `ONEWAY` computes the sums of squares for each order polynomial from weighted polynomial contrasts, using the category of the independent variable as the metric. These contrasts are orthogonal.
- With unbalanced designs and equal spacing between groups, `ONEWAY` also computes sums of squares using the unweighted polynomial contrasts. These contrasts are not orthogonal.
- The deviation sums of squares are always calculated from the weighted sums of squares[2].

Example

```
ONEWAY WELL BY EDUC6
  /POLYNOMIAL=2.
```

- `ONEWAY` requests an analysis of variance of *WELL* by *EDUC6* with second-order (quadratic) polynomial contrasts.
- The ANOVA table is expanded to include both linear and quadratic terms.

---

[2] Speed, M. F. 1976. Response curves in the one way classification with unequal numbers of observations per cell. In: *Proceedings of the Statistical Computing Section*. Alexandria, VA: American Statistical Association.

CONTRAST Subcommand

`CONTRAST` specifies a priori contrasts to be tested by the *t* statistic. The specification on `CONTRAST` is a vector of coefficients, where each coefficient corresponds to a category of the independent variable. The Contrast Coefficients table displays the specified contrasts for each group and the Contrast Tests table displays the value of the contrast and its standard error, the *t* statistic, and the degrees of freedom and two-tailed probability of *t* for each variable. Both pooled- and separate-variance estimates are displayed.

- A contrast coefficient must be specified or implied for every group defined for the independent variable. If the number of contrast values is not equal to the number of groups, the contrast test is not performed.
- The contrast coefficients for a set should sum to 0. If they do not, a warning is issued. `ONEWAY` will still give an estimate of this contrast.
- Coefficients are assigned to groups defined by ascending values of the independent variable.
- The notation *n***c* can be used to indicate that coefficient *c* is repeated *n* times.

Example

```
ONEWAY V1 BY V2
  /CONTRAST = -1 -1 1 1
  /CONTRAST = -1 0 0 1
  /CONTRAST = -1 0 .5 .5.
```

- *V2* has four levels.
- The first `CONTRAST` subcommand contrasts the combination of the first two groups with the combination of the last two groups.
- The second `CONTRAST` subcommand contrasts the first group with the last group.
- The third `CONTRAST` subcommand contrasts the first group with the combination of the third and fourth groups.

Example

```
ONEWAY V1 BY V2
  /CONTRAST = -1 1 2*0
  /CONTRAST = -1 1 0 0
  /CONTRAST = -1 1.
```

- The first two `CONTRAST` subcommands specify the same contrast coefficients for a four-group analysis. The first group is contrasted with the second group in both cases.
- The first `CONTRAST` uses the *n***c* notation.
- The last `CONTRAST` does not work because only two coefficients are specified for four groups.

POSTHOC Subcommand

`POSTHOC` produces post hoc tests for comparisons of all possible pairs of group means or multiple comparisons. In contrast to a priori analyses specified on the `CONTRAST` subcommand, post hoc analyses are usually not planned at the beginning of the study but are suggested by the data in the course of the study.

- Twenty post hoc tests are available. Some detect homogeneity subsets among the groups of means, some produce pairwise comparisons, and others perform both. `POSTHOC` produces a Multiple Comparison table showing up to 10 test categories. Nonempty group means are sorted in ascending order, with asterisks indicating significantly different groups. In addition, homogeneous subsets are calculated and displayed in the Homogeneous Subsets table if the test is designed to detect homogeneity subsets.
- When the number of valid cases in the groups varies, the harmonic mean of the group sizes is used as the sample size in the calculation for homogeneity subsets except for `QREGW` and `FREGW`. For `QREGW` and `FREGW` and tests for pairwise comparison, the sample sizes of individual groups are always used.
- You can specify only one `POSTHOC` subcommand per `ONEWAY` command. If more than one is specified, the last specification takes effect.

- You can specify one alpha value used in all `POSTHOC` tests using keyword `ALPHA`. The default is 0.05.

SNK. *Student-Newman-Keuls procedure based on the Studentized range test.* Used for detecting homogeneity subsets.

TUKEY. *Tukey's honestly significant difference.* This test uses the Studentized range statistic to make all pairwise comparisons between groups. Used for pairwise comparison and for detecting homogeneity subsets.

BTUKEY. *Tukey's b.* Multiple comparison procedure based on the average of Studentized range tests. Used for detecting homogeneity subsets.

DUNCAN. *Duncan's multiple comparison procedure based on the Studentized range test*. Used for detecting homogeneity subsets.

SCHEFFE. *Scheffé's multiple comparison* t *test.* Used for pairwise comparison and for detecting homogeneity subsets.

DUNNETT(refcat). *Dunnett's two-tailed t test.* Used for pairwise comparison. Each group is compared to a reference category. You can specify a reference category in parentheses. The default is the last category. This keyword must be spelled out in full.

DUNNETTL(refcat). *Dunnett's one-tailed t test.* Used for pairwise comparison. This test indicates whether the mean of each group (except the reference category) is *smaller* than that of the reference category. You can specify a reference category in parentheses. The default is the last category. This keyword must be spelled out in full.

DUNNETTR(refcat). *Dunnett's one-tailed t test.* Used for pairwise comparison. This test indicates whether the mean of each group (except the reference category) is *larger* than that of the reference category. You can specify a reference category in parentheses. The default is the last category. This keyword must be spelled out in full.

BONFERRONI. *Bonferroni t test.* This test is based on Student's *t* statistic and adjusts the observed significance level for the fact that multiple comparisons are made. Used for pairwise comparison.

LSD. *Least significant difference t test.* Equivalent to multiple *t* tests between all pairs of groups. Used for pairwise comparison. This test does not control the overall probability of rejecting the hypotheses that some pairs of means are different, while in fact they are equal.

SIDAK. *Sidak t test.* Used for pairwise comparison. This test provides tighter bounds than the Bonferroni test.

GT2. *Hochberg's GT2.* Used for pairwise comparison and for detecting homogeneity subsets. This test is based on the Studentized maximum modulus test. Unless the cell sizes are extremely unbalanced, this test is fairly robust even for unequal variances.

GABRIEL. *Gabriel's pairwise comparisons test based on the Studentized maximum modulus test.* Used for pairwise comparison and for detecting homogeneity subsets.

FREGW. *Ryan-Einot-Gabriel-Welsch's multiple stepdown procedure based on an* F *test.* Used for detecting homogeneity subsets.

QREGW. *Ryan-Einot-Gabriel-Welsch's multiple stepdown procedure based on the Studentized range test.* Used for detecting homogeneity subsets.

T2. *Tamhane's T2.* Used for pairwise comparison. This test is based on a *t* test and can be applied in situations where the variances are unequal.

T3. *Tamhane's T3.* Used for pairwise comparison. This test is based on the Studentized maximum modulus test and can be applied in situations where the variances are unequal.

GH. *Games and Howell's pairwise comparisons test based on the Studentized range test.* Used for pairwise comparison. This test can be applied in situations where the variances are unequal.

C. *Dunnett's* C. Used for pairwise comparison. This test is based on the weighted average of Studentized ranges and can be applied in situations where the variances are unequal.

WALLER(kratio). *Waller-Duncan* t *test.* Used for detecting homogeneity subsets. This test uses a Bayesian approach. The *k*-ratio is the Type 1/Type 2 error seriousness ratio. The default value is 100. You can specify an integer greater than 1 within parentheses.

Example

```
ONEWAY WELL BY EDUC6
  /POSTHOC=SNK SCHEFFE ALPHA=.01.
```

- `ONEWAY` requests two different post hoc tests. The first uses the Student-Newman-Keuls test and the second uses Scheffé's test. Both tests use an alpha of 0.01.

RANGES Subcommand

`RANGES` produces results for some post hoc tests. It is available only through syntax. You can always produce the same results using the `POSTHOC` subcommand.

- Up to 10 `RANGE` subcommands are allowed. The effect is cumulative. If you specify more than one alpha value for different range tests, the last specified value takes effect for all tests. The default is 0.05.
- Keyword `MODLSD` on the `RANGE` subcommand is equivalent to keyword `BONFERRONI` on the `POSTHOC` subcommand. Keyword `LSDMOD` is an alias for `MODLSD`.

PLOT MEANS Subcommand

`PLOT MEANS` produces a chart that plots the subgroup means (the means for each group defined by values of the factor variable).

STATISTICS Subcommand

By default, `ONEWAY` displays the ANOVA table showing between- and within-groups sums of squares, mean squares, degrees of freedom, *F* ratio, and probability of *F*. Use `STATISTICS` to obtain additional statistics.

BROWNFORSYTHE. *Brown-Forsythe statistic.* The Brown-Forsythe statistic, degrees of freedom, and the significance level are computed for each dependent variable.

WELCH. *Welch statistic.* The Welch statistic, degrees of freedom, and the significance level are computed for each dependent variable.

DESCRIPTIVES. *Group descriptive statistics.* The statistics include the number of cases, mean, standard deviation, standard error, minimum, maximum, and 95% confidence interval for each dependent variable for each group.

EFFECTS. *Fixed- and random-effects statistics.* The statistics include the standard deviation, standard error, and 95% confidence interval for the fixed-effects model, and the standard error, 95% confidence interval, and estimate of between-components variance for the random-effects model.

HOMOGENEITY. *Homogeneity-of-variance tests.* The statistics include Levene test statistics based on the mean, median, and trimmed mean, degrees of freedom (including adjusted degrees of freedom for the test based on the median), and the significance level displayed in the Tests of Homogeneity of Variances table.

NONE. *No optional statistics.* This is the default.

ALL. *All statistics available for* `ONEWAY`.

MISSING Subcommand

`MISSING` controls the treatment of missing values.

- Keywords `ANALYSIS` and `LISTWISE` are alternatives. Each can be used with `INCLUDE` or `EXCLUDE`. The default is `ANALYSIS` and `EXCLUDE`.
- A case outside of the range specified for the grouping variable is not used.

ANALYSIS. *Exclude cases with missing values on a pair-by-pair basis.* A case with a missing value for the dependent or grouping variable for a given analysis is not used for that analysis. This is the default.

LISTWISE. *Exclude cases with missing values listwise.* Cases with missing values for any variable named are excluded from all analyses.

EXCLUDE. *Exclude cases with user-missing values.* User-missing values are treated as missing. This is the default.

INCLUDE. *Include user-missing values.* User-missing values are treated as valid values.

MATRIX Subcommand

`MATRIX` reads and writes matrix data files.

- Either `IN` or `OUT` and a matrix file in parentheses are required.
- You cannot specify both `IN` and `OUT` on the same `ONEWAY` procedure.
- Use `MATRIX=NONE` to explicitly indicate that a matrix data file is not being written or read.

OUT ('savfile'|'dataset'). *Write a matrix data file or dataset.* Specify either a filename, a previously declared dataset name, or an asterisk, enclosed in parentheses. Filenames should be enclosed in quotes and are stored in the working directory unless a path is included as part of the file specification. If you specify an asterisk (*), the matrix data file replaces the active dataset. If you specify an asterisk or a dataset name, the file is not stored on disk unless you use SAVE or XSAVE.

IN ('savfile'|'dataset'). *Read a matrix data file or dataset.* Specify either a filename, dataset name created during the current session, or an asterisk enclosed in parentheses. An asterisk reads the matrix data from the active dataset. Filenames should be enclosed in quotes and are read from the working directory unless a path is included as part of the file specification.

Matrix Output

- `ONEWAY` writes means, standard deviations, and frequencies to a matrix data file that can be used by subsequent `ONEWAY` procedures. For a description of the file, see below.

Matrix Input

- `ONEWAY` can read the matrices it writes, and it can also read matrix materials that include the means, category frequencies, pooled variance, and degrees of freedom for the pooled variance. The pooled variance has a *ROWTYPE_* value MSE, and the vector of degrees of freedom for the pooled variance has the *ROWTYPE_* value DFE.
- The dependent variables named on `ONEWAY` can be a subset of the dependent variables in the matrix data file.
- `MATRIX=IN` cannot be specified unless an active dataset has already been defined. To read an existing matrix data file at the beginning of a session, use `GET` to retrieve the matrix file and then specify `IN(*)` on `MATRIX`.

Format of the Matrix Data File

- The matrix data file includes two special variables created by the program: *ROWTYPE_* and *VARNAME_*.
- *ROWTYPE_* is a short string variable with values MEAN, STDDEV, and N.
- *VARNAME_* is a short string variable that never has values for procedure `ONEWAY`. *VARNAME_* is included with the matrix materials so that matrices written by `ONEWAY` can be read by procedures that expect to read a *VARNAME_* variable.
- The independent variable is between variables *ROWTYPE_* and *VARNAME_*.
- The remaining variables in the matrix file are the dependent variables.

Split Files

- When split-file processing is in effect, the first variables in the matrix data file are the split variables, followed by *ROWTYPE_*, the independent variable, *VARNAME_*, and the dependent variables.

- A full set of matrix materials is written for each split-file group defined by the split variable(s).
- A split variable cannot have the same variable name as any other variable written to the matrix data file.
- If split-file processing is in effect when a matrix is written, the same split file must be in effect when that matrix is read by any procedure.
- Generally, matrix rows, independent variables, and dependent variables can be in any order in the matrix data file read by keyword `IN`. However, all split-file variables must precede variable *ROWTYPE_*, and all split-group rows must be consecutive. `ONEWAY` ignores unrecognized *ROWTYPE_* values.

Missing Values

Missing-value treatment affects the values written to an matrix data file. When reading a matrix data file, be sure to specify a missing-value treatment on `ONEWAY` that is compatible with the treatment that was in effect when the matrix materials were generated.

Example

```
GET FILE=GSS80.
ONEWAY  WELL BY EDUC6
  /MATRIX=OUT(ONEMTX).
```

- `ONEWAY` reads data from file *GSS80* and writes one set of matrix materials to the file *ONEMTX*.
- The active dataset is still *GSS80*. Subsequent commands are executed on *GSS80*.

Example

```
GET FILE=GSS80.
ONEWAY  WELL BY EDUC6
  /MATRIX=OUT(*).
LIST.
```

- `ONEWAY` writes the same matrix as in the example above. However, the matrix data file replaces the active dataset. The `LIST` command is executed on the matrix file, not on the *GSS80* file.

Example

```
GET FILE=PRSNNL.
FREQUENCIES VARIABLE=AGE.
ONEWAY  WELL BY EDUC6
 /MATRIX=IN(ONEMTX).
```

- This example performs a frequencies analysis on *PRSNNL* and then uses a different file for `ONEWAY`. The file is an existing matrix data file.
- `MATRIX=IN` specifies the matrix data file.
- *ONEMTX* does not replace *PRSNNL* as the active dataset.

Example

```
GET FILE=ONEMTX.
ONEWAY  WELL BY EDUC6
  /MATRIX=IN(*).
```

- The `GET` command retrieves the matrix data file *ONEMTX*.
- `MATRIX=IN` specifies an asterisk because the active dataset is the matrix data file *ONEMTX*. If `MATRIX=IN(ONEMTX)` is specified, the program issues an error message, since *ONEMTX* is already open.
- If the `GET` command is omitted, the program issues an error message.

TEMPLATE Subcommand

`TEMPLATE` uses an existing file as a template and applies it to the means plot (`/PLOT MEANS`), if requested, for the current `ONEWAY` command. The template overrides the default settings that are used to create the plot. Templates are created in the Chart Editor by saving an existing chart as a template.

Example

```
ONEWAY perform BY group
  /MISSING ANALYSIS
  /STATISTICS HOMOGENEITY
  /PLOT MEANS
  /TEMPLATE='/templates/mytemplate.sgt'.
```

CRITERIA Subcommand

The optional `CRIERIA` subcommand controls the significance level to estimate the confidence intervals. The **`CILEVEL`** keyword controls the significance level to estimate the confidence intervals. The specified number must be a single numerical value between 0 and 1, with 0.95 as the default setting.

Note: `CILEVEL` resets the confidence interval value for both the descriptive statistics and the effect sizes.

Example

```
ONEWAY perform BY group
  /MISSING ANALYSIS
  /STATISTICS HOMOGENEITY
  /PLOT MEANS
  /TEMPLATE='/templates/mytemplate.sgt'.
  /CRITERIA = CILEVEL(0.95)
```

ES Subcommand

The optional ES subcommand controls the effect size estimation. It is omitted by default. When specified, at least one of the following keywords must be specified to calculate the effect sizes:

OVERALL
This optional keyword controls the calculation of the effect size for the overall test. When specified, the “ANOVA Effect Sizes” table displays in the output.

CONTRAST
This optional keyword controls the calculation of the effect size for the contrast test. When specified, the “Contrast Effect Sizes” table displays in the output. **`CONTRAST(POOLED)`** uses the pooled standard deviation for all the groups as the standardizer when estimating the effect size, (the default setting). **`CONTRAST({GROUP})`** uses the pooled standard deviation for groups involved in the contrast as the standardizer. This keyword takes effect in the presence of at least one user-supplied contrast, and is ignored when no contrasts are specified.

Example

```
ONEWAY V1 BY V2
  /CONTRAST = -1 -1 1 1
  /CONTRAST = -1 0 0 1
  /CONTRAST = -1 0 .5 .5.
  /ES = CONTRAST(POOLED)
```

References

Speed, M. F. 1976. Response curves in the one way classification with unequal numbers of observations per cell. In: *Proceedings of the Statistical Computing Section*. Alexandria, VA: American Statistical Association.

OUTPUT MODIFY

Note: Square brackets used in the `OUTPUT MODIFY` syntax chart are required parts of the syntax and are not used to indicate optional elements. Any equals signs (=) displayed in the syntax chart are required. All subcommands except SELECT are optional.

```
OUTPUT MODIFY NAME={*   }
                   {ALL }
                   {name}

 /REPORT PRINTREPORT = {NO**  }
                       {YES   }
```

```
/SELECT CHARTS HEADINGS LOGS MODELS
        TABLES TEXTS TREES WARNINGS
        OUTLINEHEADERS UNKNOWN OTHER
        PAGETITLE NOTES
or
/SELECT ALL EXCEPT = [list]

/IF  COMMANDS = ["expression", "expression(integer)"
                 "expression(LAST)", LAST]
     SUBTYPES = ["expression" "expression"...]
     LABELS = [EXACT("expression") BEGINS("expression")
               ENDS("expression") CONTAINS("expression")]
     INSTANCES = [n n... LAST]
     VISIBLEONLY={NO**}
                 {YES }

/DELETEOBJECT DELETE={NO**}
                     {YES }

/INDEXING
 INDEXTYPE = {NUMBER** }
             {LETTERLOW}
             {LETTERCAP}
             {ROMANLOW }
             {ROMANCAP }
 INDEXSTART = {1**     }
              {integer }
              {"letter"}
              {"roman" }

/OBJECTPROPERTIES
 VISIBLE = {ASIS**}
           {YES   }
           {NO    }
 OUTLINELABEL = "expression"
 SIZE = {PERCENT    }  ({width, height       })
        {CENTIMETERS}   {PROPORTIONAL, height}
        {INCHES     }   {width, PROPORTIONAL }
        {POINTS     }

/TABLE
 TABLETITLE = "expression"
 TRANSPOSE = {NO**}
             {YES }
 TLOOK = {"filespec"}
         {"name"    }
         {NONE      }
 SELECTEDLAYER = [{"variable", "category"}]
                  {"variable", number    }
                  {number, "category"    }
                  {number, number        }
 SORT = {COLLABEL("label")  }
      {COLPOSITION(number)}
 SORTDIRECTION = {ASCENDING**}
                 {DESCENDING }
 TABLESUMMARY = "expression"
 PIVOT = [expression]

/TABLECELLS
 SELECT = ["expression", COUNT, MEAN, MEDIAN, RESIDUAL,
           PERCENT, SIGNIFICANCE, POSITION(integer),
           BODY, HEADERS, FOOTNOTES, TITLE, CAPTION, …]
 SELECTDIMENSION = {BOTH** }
                   {COLUMNS}
                   {ROWS   }
 SELECTCONDITION = {ALL**,                                             }
                   {["expression", TOP(integer), BOTTOM(integer),
                     PARENT(string), CHILD(string),…]}
 COLUMNWIDTH = {CENTIMETERS(value)}
               {INCHES(value)     }
               {POINTS(value)     }
 APPLYTO = {CELL**      }
           {COLUMN      }
           {ROW         }
           (COLUMNDATA  }
           {ROWDATA     }
           {COLUMNHEADER}
           {ROWHEADER   }
 REVERTTODEFAULT = {NO**}
                   {YES }
 FORMAT=format
 FONT = "fontfamily"
 FONTSIZE = value
 STYLE = REGULAR BOLD ITALIC UNDERLINE
 TEXTCOLOR = {color       }
             {RGB(r, g, b)}
 BACKGROUNDCOLOR = {color       }
                   {RGB(r, g, b)}
 REPLACE = {"expression"}
           {number      }
 HIDE = {NO**}
        {YES }
 UNGROUP = {NO**}
           {YES }

/GRAPHS
 CTEMPLATE = {["filespec", "filespec", ...]}
             {[name, name, ...]           }
 VIZSTYLESHEET = "stylesheet name"

/TEXTS
 TEXT = "expression"
 FONT = "fontfamily"
 FONTSIZE = value
 STYLE = REGULAR BOLD ITALIC UNDERLINE
 TEXTCOLOR = {color       }
             {RGB(r, g, b)}
```

- If included, the REPORT subcommand must be the first subcommand.
- Multiple blocks of output modifications are allowed. Each block of modification specifications must begin with a SELECT subcommand.
- Valid color values for TEXTCOLOR and BACKGROUNDCOLOR are `BLACK, BLUE, CYAN, DARKGRAY, GRAY, GREEN, LIGHTGRAY, MAGENTA, ORANGE, PINK, RED, WHITE, YELLOW`.

This command takes effect immediately. It does not read the active dataset or execute pending transformations. See the topic Command Order for more information.

Release History

Release 27.0

- Added support for the **PIVOT** keyword in the **TABLES** subcommand.
- Added support for the **HIDE** and **UNGROUP** keywords in the **TABLECELLS** subcommand.
- Added support for the **PARENT** and **CHILD** options for the **SELECTCONDITION** keyword in the **TABLECELLS** subcommand.

Release 22.0

- Command introduced.

Example

```
OUTPUT MODIFY
 /SELECT TABLES
 /IF SUBTYPES=["Frequencies"]
 /TABLECELLS SELECT=["Total"]
  SELECTDIMENSION=ROWS
  STYLE=BOLD APPLYTO=ROW.
```

Overview

The `OUTPUT MODIFY` command applies formatting and other changes to the contents of the active Viewer window. Changes that can be applied include:

- All or selected viewer objects
- Selected types of output objects (for example, charts, logs, pivot tables)
- Pivot table content based on conditional expressions
- Outline (navigation) pane content

The types of changes you can make include:

- Delete objects
- Index objects (add a sequential numbering scheme)
- Change the visible property of objects
- Change the outline label text
- Transpose rows and columns in pivot tables
- Change the selected layer of pivot tables
- Change the formatting of selected areas or specific cells in a pivot table based on conditional expressions (for example, make all significance values less than 0.05 bold)

Basic Specification

The basic specification is the command name followed by a SELECT subcommand that contains and one or more subcommands that specify how you want to change the output. For example:

```
OUTPUT MODIFY
  /SELECT TABLES
  /IF SUBTYPES=["Frequencies"]
  /TABLECELLS SELECT=["Total"]
    SELECTDIMENSION=ROWS
    STYLE=BOLD APPLYTO=ROW.
```

- The `SELECT` subcommand specifies that the command applies only to pivot tables.
- The `IF` subcommand specifies that the command applies only to the table subtype "Frequencies".
- The `TABLECELLS` subcommand specifies that the font style of the row that contains the text "Total" should be bold.

Syntax Rules

- All subcommands except `SELECT` are optional.
- If included, the `REPORT` subcommand must be the first subcommand.
- Multiple blocks of specifications for different conditions can be specified. Each block must begin with a `SELECT` subcommand. For more information, see the "Basic Operations" topic.
- Subcommands other than `SELECT` can appear in any order.

Basic Operation

- `OUTPUT MODIFY` operates on all applicable output objects that exist in the active Viewer before the `OUTPUT MODIFY` command runs.
- `OUTPUT MODIFY`must come after any commands that produce output to which you want `OUTPUT MODIFY` to apply.
- The effects of multiple `SELECT` blocks within a single `OUTPUT MODIFY` command and the effects of multiple `OUTPUT MODIFY` commands are both cumulative. The main difference between these approaches is that subsequent `OUTPUT MODIFY` commands can apply to output objects created by previous `OUTPUT MODIFY` commands. For example, if the first `OUTPUT MODIFY` command includes a `REPORT` subcommand, and a second `OUTPUT MODIFY` command that immediately follows it includes `IF COMMANDS=[LAST]`, the changes that are specified on the second `OUTPUT MODIFY` command are applied to the report output created by the first `OUTPUT MODIFY` command, not the last command before both `OUTPUT MODIFY` commands.

NAME Keyword

The optional `NAME` keyword specifies the Viewer document you want to modify. This is useful when more than one Viewer document is open and you want to ensure that the command operates on the correct document.

NAME=*
The currently active output document.

NAME=ALL
All open output documents.

NAME=name
This is the name assigned by a previous `OUTPUT NAME`, `OUTPUT OPEN`, `OUTPUT NEW` command, or automatically assigned when the document is created or opened in the current session; it is not the file name associated with a saved output document. In the Viewer window title bar, the name is enclosed in square brackets.

SELECT Subcommand

The `SELECT` subcommand specifies the types of output objects that you want to modify.

CHARTS
Chart objects. This object type includes charts that are created by commands such as `GRAPH` and `GGRAPH` and charts that are created by statistical procedures (for example, the `BARCHART` subcommand of the `FREQUENCIES` command). It does not include tree diagrams that are produced by the `TREE` procedure or charts in models that are displayed in the Model Viewer.

HEADINGS
Text objects that are labeled "Title" in the outline pane of the Viewer.

LOGS

Log text objects. Log objects contain certain types of error and warning messages. With SET `PRINTBACK=ON`, log objects also contain the command syntax run during the session. Log objects are labeled "Log" in the outline pane of the Viewer.

MODELS

Output objects that are displayed in the Model Viewer. A single model object can contain multiple views of the model, including both tables and charts.

TABLES

Output objects that are pivot tables in the Viewer. This category does not include Notes tables or tables displayed in model viewers.

TEXTS

Text objects that are not logs or headings. This category includes objects that are labeled "Text Output" in the outline pane of the Viewer.

TREES

Tree model diagrams that are produced by the TREE procedure (Forecasting and Decision Trees).

WARNINGS

Warnings objects. Warnings objects contain certain types of error and warning messages.

OUTLINEHEADERS

Headers in the outline pane that have no corresponding object in the contents pane. For example, output objects that are created by a statistical procedure are nested under the procedure name in the outline pane. The procedure name is the header that has no corresponding object in the contents pane.

PAGETITLE

Page titles that are created by the `TITLE` or `SUBTITLE` command.

NOTES

Notes tables that are created by statistical and charting procedures. The Notes table contains information about the dataset that was used, missing values, the command syntax that was used to run the procedure, and the computer resources that were used to run the procedure.

UNKNOWN

IBM SPSS Statistics objects that do not have a known type. If new object types are available in future releases, these objects will be treated as "unknown" in earlier releases.

OTHER

Objects that are created in other applications and pasted into the Viewer, such as images.

ALL

All output objects. If you specify ALL, any other output type keywords are ignored. You can use EXCEPT with ALL to exclude specific output types. The exception list must be enclosed in parentheses.

Example

```
OUTPUT MODIFY
  /SELECT ALL EXCEPT (TABLES)
  /DELETEOBJECT DELETE = YES.
```

In this example, all output objects except pivot tables are deleted from the Viewer.

IF Subcommand

The `IF` subcommand specifies particular output objects of the types determined by `SELECT`. Without an `IF` subcommand, all objects of the specified types are selected. If you specify multiple conditions, only those objects that meet all conditions are selected.

COMMANDS Keyword

The `COMMANDS` keyword restricts the selection to the specified commands. The keyword `COMMANDS` must be followed by an equals sign (=) and a list of quoted command identifiers enclosed in square bracket, as in:

```
OUTPUT MODIFY
 /SELECT TABLES
 /IF COMMANDS = ['Frequencies' 'Factor Analysis']
  ...
```

The command identifier can be followed by a value in parentheses. The value can be an integer that indicates the instance of the command or the keyword `LAST`. The keyword `LAST` can be used by itself to specify the output of the last command before `OUTPUT MODIFY`. For example:

```
OUTPUT MODIFY
  /SELECT TABLES
  /IF COMMANDS=['Frequencies(2)', 'Descriptives(LAST)', LAST]
  ...
```

In this example, the specified changes are applied to:

- The second instance of the `FREQUENCIES` command.
- The last instance of the `DESCRIPTIVES` command.
- The last command before the `OUTPUT MODIFY` command, if that command contains pivot table output.

Command identifiers are:

- Unique. No two commands have the same identifier.
- Not case-sensitive.
- Not subject to translation, which means they are the same for all language versions and output languages.
- Often not the same or even similar to the command name. You can obtain the identifier for a particular command by generating output from the command in the Viewer and then right-clicking the command heading in the outline pane and selecting **Copy OMS Command Identifier** from the menu.

Command identifiers are available for all statistical and charting procedures and any other commands that produce blocks of output with their own identifiable heading in the outline pane of the Viewer. For example, `CASESTOVARS` and `VARSTOCASES` have corresponding identifiers ('Cases to Variables' and 'Variables to Cases') because they produce their own output blocks (with command headings in the outline pane that happen to match the identifiers), but `FLIP` does not because any output produced by `FLIP` is included in a generic Log text object.

SUBTYPES Keyword

The `SUBTYPES` keyword restricts the selection to the specified table types The keyword `SUBTYPES` must be followed by an equals sign (=) and a list of quoted subtype identifiers enclosed in square bracket, as in:

```
OUTPUT MODIFY
 /SELECT TABLES
 /IF SUBTYPES = ['Descriptive Statistics' 'Coefficients']
 ...
```

- Subtypes apply only to tables that would be displayed as pivot tables in the Viewer.
- Like command identifiers, subtype identifiers are not case-sensitive and are not subject to translation.
- Unlike command identifiers, subtype identifiers are not necessarily unique. For example, multiple commands produce a table with the subtype identifier "Descriptive Statistics," but not all of those tables share the same structure. If you want only a particular table type for a particular command, use both the `COMMANDS` and `SUBTYPES` keywords.
- You can obtain the identifier for a particular table by generating output from the command in the Viewer and then right-clicking outline item for the Table in the outline pane of the Viewer and selecting **Copy OMS Table Subtype** from the menu. The identifiers are generally fairly descriptive of the particular table type.

LABELS Keyword

The `LABELS` keyword selects particular output objects according to the text displayed in the outline pane of the Viewer. The keyword `LABELS` must be followed by an equals sign (=) and a list of expressions enclosed in square brackets, as in:

```
OUTPUT MODiFY
 /SELECT TABLES
 /IF LABELS = [EXACT('Age'), CONTAINS('Crosstabulation')]
 ...
```

- Each expression must start with a keyword, followed by a quoted value enclosed in parentheses.
- Valid keywords are `EXACT`, `CONTAINS`, `BEGINS`, and `ENDS`.
- Label values are case-sensitive.

The `LABELS` keyword is useful for differentiating between multiple graphs or multiple tables of the same type in which the outline text reflects some attribute of the particular output object such as the variable names or labels. There are, however, a number of factors that can affect the label text:

- If split file processing is on, split file group identification is often appended to the label.
- Labels that include information about variables or values are affected by the `OVARS` and `ONUMBERS` settings on the SET command.
- Labels are affected by the current output language setting (`SET OLANG`).

INSTANCES Keyword

The `INSTANCES` subcommand selects the *n*th instance of an object that matches the other criteria on the `IF` subcommand within a single instance of a command. The keyword `INSTANCES` must be followed by an equals sign (=) and a list of positive integers or the keyword LAST enclosed in square brackets.

Example

```
OUTPUT MODIFY
 /SELECT TABLES
 /IF COMMANDS = ['Frequencies']
     SUBTYPES = ['Frequencies']
     INSTANCES = [1 LAST]
 ...
OUTPUT MODIFY
 /SELECT TABLES
 /IF COMMANDS = ['Frequencies']
     INSTANCES = [1 LAST]
 ...
```

- The first command selects the first and last frequency tables from each `FREQUENCIES` command.
- The second command, in the absence of a `SUBTYPES` or `LABELS` specification, selects the first and last tables of any kind from the selected command.

VISIBLEONLY Keyword

The `VISIBLEONLY` keyword indicates if the changes should be applied to all output objects that meet the selection criteria or only those objects that meet the criteria and are currently visible in the Viewer document (objects can be present but hidden).

VISIBLEONLY=NO
Changes are applied to both visible and hidden objects.

VISIBLE=YES
Changes are applied only to visible objects.

DELETEOBJECT Subcommand

The `DELETEOBJECT` subcommand deletes all objects that meet the selection criteria.

DELETE=NO
The objects that meet the selection criteria are not deleted. This option is the default.

DELETE=YES
The objects that meet the selection criteria are deleted.

Example

```
OUTPUT MODIFY
  /SELECT ALL EXCEPT (TABLES)
  /DELETEOBJECT DELETE = YES.
```

In this example, all output objects except pivot tables are deleted from the Viewer.

INDEXING Subcommand

The `INDEXNG` subcommand adds a sequential number, letter, or roman numeral to the objects that meet the selection criteria. Index values are used by the `OBJECTPROPERTIES` command (`OUTLINELABEL` keyword) and `TABLE` subcommand (`TABLETITLE` keyword).

Example

```
OUTPUT MODIFY
  /SELECT ALL
  /INDEXING
   INDEXTYPE=LETTERLOW
   INDEXSTART='a'
  /OBJECTPROPERTIES
   OUTLINELABEL = ")INDEX - )LABEL".
```

INDEXTYPE Keyword

The `INDEXTYPE` keyword indicates whether numbers, letters, or roman numerals are used for the index values.

NUMBER
Integers are used to index the objects. This option is the default.

LETTERLOW
Lowercase letters are used to index the objects.

LETTERCAP
Uppercase letters are used to index the objects.

ROMANLOW
Lowercase roman numerals are used to index the objects.

ROMANCAP
Uppercase roman numerals are used to index the objects.

INDEXSTART Keyword

The `INDEXSTART` keyword specifies the starting index value.

- For `INDEXTYPE=NUMBER`, the value of `INDEXSTART` must be an integer. The default is 1.
- For `INDEXTYPE=LETTERLOW`, the value of `INDEXSTART` must be a quoted string of one or more lowercase letters. The default is ’a’.
- For `INDEXTYPE=LETTERCAP`, the value of `INDEXSTART` must be a quoted string of one or more uppercase letters. The default is ’A’.
- For `INDEXTYPE=ROMANLOW`, the value of `INDEXSTART` must be a quoted string of lowercase letters that represents a valid number that is expressed in roman numerals. The default is ’i’.
- For `INDEXTYPE=ROMANCAP`, the value of `INDEXSTART` must be a quoted string of uppercase letters that represents a valid number that is expressed in roman numerals. The default is ’I’.

OBJECTPROPERTIES Subcommand

The `OBJECTPROPERTIES` subcommand modifies the visible property and outline text of all objects that meet the selection criteria. It also modifies the size of charts and tree diagrams.

VISIBLE Keyword

The `VISIBLE` keyword determines whether the objects that meet the selection criteria are visible or hidden in the Viewer. (Hidden object can be shown by double-clicking the icon next to the object in the outline pane.)

ASIS
Do not change the visible property of the objects. This option is the default.

YES
Make all objects that meet the selection criteria visible.

NO
Hide all objects that meet the selection criteria.

Example

```
OUTPUT MODIFY
  /SELECT ALL EXCEPT (TABLES)
  /OBJECTPROPERTIES VISIBLE=NO.
```

In this example, all output objects except pivot tables are hidden in the Viewer.

OUTLINELABEL Keyword

The `OUTLINELABEL` keyword modifies the text of outline labels in the outline pane. The keyword is followed by an equals sign and a quoted value. You can also include special variables to insert date, time, and index values.

)DATE
Current date in the form dd-mmm-yyyy.

)ADATE
Current date in the form mm/dd/yyyy.

)SDATE
Current date in the form yyyy/mm/dd.

)EDATE
Current date in the form dd.mm.yyyy.

)TIME
Current 12-hour clock time in the form hh:mm:ss.

)ETIME
Current 24-hour clock time in the form hh:mm:ss.

)INDEX
The index as specified on the `INDEXING` subcommand.

)LABEL
The current label text. This setting allows you to add other text or special variable values to the existing label.

Example

```
OUTPUT MODIFY
  /SELECT TABLES
  /INDEXING
   INDEXTYPE=NUMBER INDEXSTART=1
  /OBJECTPROPERTIES
   OUTLINELABEL = ")INDEX - )EDATE - )LABEL".
```

In this example, the outline labels for all pivot tables are modified to insert a sequential index number and the current date, separated by dashes, before the current label.

SIZE Keyword

The `SIZE` keyword modifies the size of charts, tree diagrams, and objects that are created by other applications and pasted into the Viewer document. The keyword is followed by an equals sign, a keyword that specifies measurement unit, and width and height values that are enclosed in parentheses.

- Valid keywords for measurement unit are `PERCENT`, `CENTIMETERS`, `INCHES`, and `POINTS`..
- The keyword `PROPORTIONAL` can be used instead of a width or height value. This keyword maintains the aspect ratio of the original image.

Example

```
OUTPUT MODIFY
  /SELECT CHARTS
  /OBJECTPROPERTIES
   SIZE=PERCENT(50,PROPORTIONAL).
```

In this example, the width of all charts is set to 50% of the original width. The keyword PROPORTIONAL sets the height to 50%, which preserves the original aspect ratio.

TABLE Subcommand

The `TABLE` subcommand modifies properties of pivot tables, notes tables, and warning objects in Viewer (`TABLES`, `NOTES`, and `WARNINGS` keywords on the `SELECT` subcommand).

TABLETITLE keyword

The `TABLETITLE` keyword modifies the text of title text in the contents pane of the Viewer. The keyword is followed by an equals sign and a quoted value. You can also include special variables to insert date, time, and index values.

)DATE
Current date in the form dd-mmm-yyyy.

)ADATE
Current date in the form mm/dd/yyyy.

)SDATE
Current date in the form yyyy/mm/dd.

)EDATE
Current date in the form dd.mm.yyyy.

)TIME
Current 12-hour clock time in the form hh:mm:ss.

)ETIME
Current 24-hour clock time in the form hh:mm:ss.

)INDEX
The index as specified on the `INDEXING` subcommand.

)LABEL
The current label text. This setting allows you to add other text or special variable values to the existing label.

Example

```
OUTPUT MODIFY
  /SELECT TABLES
  /INDEXING
   INDEXTYPE=NUMBER INDEXSTART=1
```

```
   /TABLE
    TABLETITLE = ")INDEX - )EDATE - )LABEL".
```

In this example, the titles for all pivot tables are modified to insert a sequential index number and the current date, separated by dashes, before the current label.

TRANSPOSE keyword

The `TRANSPOSE` keyword transposes rows and columns in pivot tables.

NO
Do not transpose rows and columns. This option is the default.

YES
Transpose rows and columns.

TLOOK keyword

The `TLOOK` keyword applies the specified TableLook to pivot tables, notes tables, and warning objects.

TLOOK=’filespec’
Applies the TableLook defined in the specified STT file. The file specification takes the general form `’/path/filename.stt’`.

TLOOK=’name’
Applies a TableLook from the `Looks` directory of the application installation directory. The value of ’name’ is the file name without the .stt extension.

TLOOK=NONE
Applies the default TableLook.

SORT and SORTDIRECTION keywords

The `SORT` and `SORTDIRECTION` keywords sort the contents of pivot tables based on the values in the specified column.

Note: These keywords work only for tables with a simple row dimension structure. These keywords are ignored if there are nested row elements.

SORT=COLLABEL(’label text’)
Sort based on the values in the column with the specified label.

SORT=COLPOSITION(integer)
Sort based on the values in the *n*th column. Column numbers start with the first row label column. For example, if the table has two row label columns, then the first data column is column 3.

SORTDIRECTION=ASCENDING
Sort in ascending order of values in the specified column. This option is the default.

SORTDIRECTION=DESCENDING
Sort in descending order of values in the specified column.

Example

```
OUTPUT MODIFY
   /SELECT TABLES
   /TABLE  SORT=COLLABEL("Percent")
    SORTDIRECTION=DESCENDING.
```

In this example, pivot tables that have a column labeled "Percent" will be sorted in descending order of the values in that column.

SELECTEDLAYER keyword

The `SELECTEDLAYER` keyword changes the displayed layer of pivot table that contains more than one layer. The keyword is followed by an equals sign and a pair of values enclosed in square brackets.

- The first value identifies the layer dimension. The value can be a quoted string that contains the label of the dimension as displayed in the table or an integer that represents the numeric position of the layer. The top layer dimension is 1.
- The second value identifies the category within the layer dimension. The value can be a quoted string that contains the label of the category as displayed in the table or an integer that represents the numeric position of the category within the layer dimension.
- For tables with multiple layer dimensions, you can specify multiple, dimension and category values.

Example

```
OUTPUT MODIFY
  /SELECT TABLES
  /TABLE SELECTLAYER=["Marital status", "Married", 2, 3].
```

- For any table with a layer dimension with the layer "Marital status", the category with the label "Married" will be displayed, if that category is present.
- For any table with at least two layer dimensions, the third category of the second layer dimension will be displayed, if that layer dimension has at least three categories.

TABLESUMMARY keyword

The `TABLESUMMARY` keyword adds comment text to the table.

- Comment text is displayed in a tooltip when you hover over a table in the Viewer.
- Screen readers read the comment text when the table has focus.
- The tooltip in the Viewer displays only the first 200 characters of the comment, but screen readers read the entire text.
- When you export output to HTML or a web report, the comment text is used as alt text.

The keyword is followed by an equals sign and a quoted value. You can also include special variables to insert date, time, and other values.

)DATE
Current date in the form dd-mmm-yyyy.

)ADATE
Current date in the form mm/dd/yyyy.

)SDATE
Current date in the form yyyy/mm/dd.

)EDATE
Current date in the form dd.mm.yyyy.

)TIME
Current 12-hour clock time in the form hh:mm:ss.

)ETIME
Current 24-hour clock time in the form hh:mm:ss.

)INDEX
The index as specified on the `INDEXING` subcommand.

)TITLE
The text of the outline label for the table.

)PROCEDURE
The name of the procedure that created the table.

)DATASET
The name of the dataset used to create the table.

\n
Inserts a line break.

PIVOT keyword

The `PIVOT` keyword pivots the specified row dimension to the specified column dimension. Any existing column dimensions are incremented outwards. For example, `PIVOT=[R1C1]` will pivot the first row dimension (the outer-most row dimension) to be the first column dimension. If there are any existing column dimensions, they will be moved outwards so that the old `C1` becomes `C2`. The keyword allows table output to adhere to APA style guidelines.

TABLECELLS Subcommand

The `TABLECELLS` subcommand specifies areas of tables to change and conditions under which to make those changes.

SELECT keyword

The `SELECT` keyword specifies the area of the table. The keyword is followed by an equals sign and a list of keywords or quoted strings, which are enclosed in square brackets.

The following keywords apply to general areas of the table:

BODY
The data cells of the tables. The body of the table does not include row and column labels.

HEADERS
All row and column labels. This option includes all labels for nested rows and columns.

FOOTNOTES
All footnote text.

TITLE
The title that is displayed above the table.

CAPTION
All caption text.

The following keywords apply to specific conditions:

COUNT
Rows or columns with any of these labels or the equivalent in the current output language: "Frequency", "Count", "N".

MEAN
Rows or columns with the label "Mean" or the equivalent in the current output language.

MEDIAN
Rows or columns with the label "Median" or the equivalent in the current output language.

PERCENT
Rows or columns with the label "Percent" or the equivalent in the current output language.

RESIDUAL
Rows or columns with any of these labels or the equivalent in the current output language: "Resid", "Residual", "Std. Residual".

SIGNIFICANCE
Rows or columns with any of these labels or the equivalents in the current output language: "Approx Sig.", "Asymp. Sig (2-sided)", "Exact Sig.", "Exact Sig. (1-sided)", "Exact Sig. (2-sided)", "Sig.", "Sig. (1-tailed)", "Sig. (2-tailed)"

CORRELATION
Rows or columns with any of these labels or the equivalent in the current output language: "Correlation Coefficient", "Correlations", "Pearson Correlation".

POSITION(integer)
Data row or column position. (The `SELECTDIMENSION` keyword specifies either rows of columns.) This does not include row label columns or column label rows.

"expression"
A quoted string that represents the value of a row or column label.

SELECTDIMENSION keyword

The `SELECTDIMENSION` keyword specifies whether `SELECT` applies to columns or rows or both.

- The keyword is followed by an equals sign and the keyword `COLUMNS`, `ROWS`, or `BOTH`. The default option is `BOTH`.
- The keyword is ignored for `SELECT` conditions `BODY`, `HEADERS`, `FOOTERNOTES`, `TITLE`, and `CAPTION`.

SELECTCONDITION keyword

The `SELECTCONDITION` keyword specifies the conditions under which to apply the changes within the area of the table specified by `SELECT` and `SELECTDIMENSION`. The keyword is followed by an equals sign and a list of keywords or quoted strings, which are enclosed in square brackets.

ALL
All conditions. This option is the default. If `ALL` is specified with any other conditions, the other conditions are ignored.

TOP(integer)
The *n* highest values in the specified area.

BOTTOM(integer)
The *n* lowest values in the specified area.

PARENT(string)
The primary string value in the specified area.

CHILD(string)
The secondary string value in the specified area.

"expression"
A quoted string that contains a comparison expression.

- The expression can include standard comparison operators (<, <=, >, >=, <>) and the keyword `ABS` for absolute value, followed by an expression in parentheses.
- The keyword `SYSMIS` indicates the system-missing value (for numeric values only).
- A lower case "x" is used as a stand-in variable to represent all values in the area of the table specified by the `SELECT` and `SELECTDIMENSION` keywords. For example: "x<=.05".
- If the expression contains string values, those values must be quoted.

COLUMNWIDTH keyword

The `COLUMNWIDTH` keyword specifies the width of the column or columns in the specified area of the table. The keyword is followed by an equals sign and the keyword `CENTIMETERS`, `INCHES`, or `POINTS`, followed by a positive number that is enclosed in parentheses.

APPLYTO keyword

The `APPLYTO` Keyword specifies what areas of the table to change based on the selection criteria.

- `APPLYTO` can apply changes to elements of the table that are not included in the selection criteria. For example, if the selection criteria specifies specific values in a specific column, `APPLYTO=ROW` applies the changes to entire rows that contain the column values that meet the selection criteria.
- `APPLYTO` is ignored for `BODY`, `HEADERS`, `FOOTNOTE`, `TITLE`, and `CAPTION`.

CELL
Applies changes only to data cells that meet the selection criteria.

COLUMN
Applies changes to entire columns that contain data values or an innermost column label that meet the selection criteria.

ROW
Applies changes to entire rows that contain data values or an innermost row label that meet the selection criteria.

COLUMNDATA
Applies changes to all data cells in the columns that contain data values that meet the selection criteria.

ROWDATA
Applies changes to all data cells in the rows that contain data values that meet the selection criteria.

COLUMNHEADER
Applies changes to the innermost column labels that meet the selection criteria.

ROWHEADER
Applies changes to the innermost row labels that meet the selection criteria.

REVERTTODEFAULT keyword

The `REVERTTODEFAULT` keyword removes any special formats or styles that have been applied to a table and reverts to the original Tablelook style.

- The keyword is followed by an equals sign and the keyword YES or `NO`. The default option is `NO`.
- When this keyword is used with other `TABLECELLS` format or style changes, `REVERTTODEFAULT` will undo any changes that are specified after it. Specify `REVERTODEFAULT` first if you want to make other changes after reverting to the default TableLook.

FORMAT keyword

The `FORMAT` keyword specifies the format for data values. The keyword is followed by an equals sign and a valid format specification, such as `F10.4`, `PCT5.1`, or `A20`. For detailed information about available formats and specifications, see Variable Types and Formats

FONT keyword

The `FONT` keyword specifies the font to use.

- The keyword is followed by an equals sign and the name of a font that is enclosed in quotation marks.
- If the specified font is not available, this setting is ignored.

FONTSIZE keyword

The `FONTSIZE` keyword specifies the font size (in points). The keyword is followed by an equals sign and a positive number.

STYLE keyword

The `STYLE` keyword specifies the font style.

- The keyword is followed by an equals sign and a list of styles.
- Valid style values are `REGULAR`, `BOLD`, `ITALIC`, and `UNDERLINE`.
- Style specifications are cumulative. If the text is already italic, the `BOLD` keyword makes the text italic and bold.
- The `REGULAR` keyword removes any special style characteristics. When used with other style keywords, it must be specified first; otherwise any previous style specifications are removed. For example, `STYLE=REGULAR BOLD` results in bold text without any other special style characteristics, but `STYLE=BOLD REGULAR` results in text that is not bold.

TEXTCOLOR and BACKGROUNDCOLOR keywords

The TEXTCOLOR keyword specifies the text color. The BACKGROUNDCOLOR keyword specifies the background color.

- The keyword is followed by an equals sign and a color keyword or RGB`(r, g, b)`, where r, g, and b are numeric red, green, and blue color values. The numeric values are integers between 1 and 255.
- Valid color keywords are `BLACK, BLUE, CYAN, DARKGRAY, GRAY, GREEN, LIGHTGRAY, MAGENTA, ORANGE, PINK, RED, WHITE, YELLOW`.

REPLACE keyword

The REPLACE keyword replaces any text that meets the selection criteria with the specified number or quoted string value.

HIDE keyword

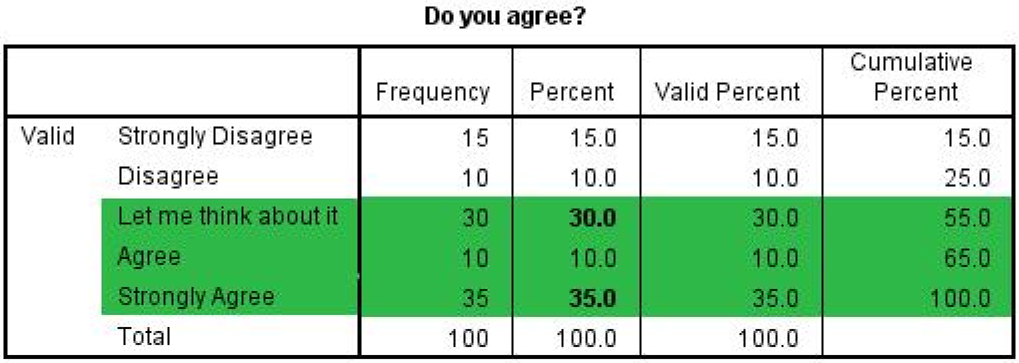
When set to YES, the HIDE keyword suppresses the selected row or column. The keyword allows table output to adhere to APA style guidelines.

UNGROUP keyword

When set to YES, the UNGROUP keyword deletes the selected row or column group header. The keyword allows table output to adhere to APA style guidelines.

Example

```
OUTPUT MODIFY
  /SELECT TABLES /IF SUBTYPES =['Frequencies']
  /TABLECELLS  SELECT=["Percent"]
   SELECTDIMENSION=COLUMNS SELECTCONDITION=["x>=30"]
   APPLYTO=CELL STYLE=REGULAR BOLD
  /SELECT TABLES /IF SUBTYPES=['Frequencies']
  /TABLECELLS SELECT=["Cumulative Percent"]
   SELECTDIMENSION=COLUMNS SELECTCONDITION=["x>=50"]
   APPLYTO=ROW BACKGROUNDCOLOR=GREEN
  /SELECT TABLES /IF SUBTYPES=["Frequencies"]
  /TABLECELLS SELECT=["Total"]
   SELECTDIMENSION=ROWS APPLYTO=ROW
   REVERTTODEFAULT=YES.
```



Do you agree?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	15	15.0	15.0	15.0
	Disagree	10	10.0	10.0	25.0
	Let me think about it	30	**30.0**	30.0	55.0
	Agree	10	10.0	10.0	65.0
	Strongly Agree	35	**35.0**	35.0	100.0
	Total	100	100.0	100.0	

Figure 1. Table modified by TABLECELLS subcommands

- The TABLESCELLS subcommand in the first SELECT block finds values greater than or equal to 30 in the column labeled "Percent" and makes them bold.

- The `TABLECELLS` subcommand in the second `SELECT` block finds values greater than or equal to 50 in the column labeled "Cumulative Percent" and makes the background color of those rows green.
- Since the first two `TABLECELLS` subcommands also change the total row at the bottom of the table, the last `TABLECELLS` subcommand uses `REVERTTODEFAULT` to remove any special formatting that was applied to the total row.

GRAPHS Subcommand

The `GRAPHS` subcommand applies a template or style sheet to charts that meet the selection criteria.

CTEMPLATE=["filespec", "filespec"...] | [name, name, ...]
Applies the specified chart template.

- Chart templates can be applied to most charts, except charts that are created with the Graphboard Template Chooser (`GGRAPH` command,`VIZSTYLESHEET` keyword).
- `CTEMPLATE=["filespec"]` applies the chart template that is defined in the specified SGT file. The file specification takes the general form `'/path/filename.sgt'`
- `CTEMPLATE=[name]` applies a chart template from the `Looks` directory of the application installation directory. The value of 'name' is the file name without the .sgt extension.
- You can specify more than one chart template. Template attributes are applied in the order the templates are specified.

VIZSTYLESHEET=name
Applies the specified style sheet.

- Style sheets apply only to charts created with the Graphboard Template Chooser (`GGRAPH` command,`VIZSTYLESHEET` keyword).
- For a list of valid names, go to the Options tab of the Graphboard Template Chooser and click **Select**.

TEXTS Subcommand

The `TEXTS` subcommand applies changes to text objects that meet the selection criteria. This includes `HEADINGS`, `LOGS`, `TEXTS`, `WARNINGS`, and `PAGETITLES` specified on the `SELECT` subcommand.

TEXT Keyword

The `TEXT` keyword modifies the text of text objects. The keyword is followed by an equals sign and a quoted value. You can also include special variables to insert date, time, and index values.

)DATE
Current date in the form dd-mmm-yyyy.

)ADATE
Current date in the form mm/dd/yyyy.

)SDATE
Current date in the form yyyy/mm/dd.

)EDATE
Current date in the form dd.mm.yyyy.

)TIME
Current 12-hour clock time in the form hh:mm:ss.

)ETIME
Current 24-hour clock time in the form hh:mm:ss.

)INDEX
The index as specified on the `INDEXING` subcommand.

)LABEL
The current text of the text object. This setting allows you to add other text or special variable values to the existing text. Without `)LABEL`, the original text is overwritten.

Example

```
OUTPUT MODIFY
  /SELECT LOGS
   TEXT=")EDATE  )LABEL".
```

In this example, the current date is added to all log objects.

FONT Keyword

The `FONT` keyword specifies the font to use.

- The keyword is followed by an equals sign and the name of a font that is enclosed in quotation marks.
- If the specified font is not available, this setting is ignored.

FONTSIZE Keyword

The `FONTSIZE` keyword specifies the font size (in points). The keyword is followed by an equals sign and a positive number.

STYLE Keyword

The `STYLE` keyword specifies the font style.

- The keyword is followed by an equals sign and a list of styles.
- Valid style values are `REGULAR`, `BOLD`, `ITALIC`, and `UNDERLINE`.
- Style specifications are cumulative. If the text is already italic, the `BOLD` keyword makes the text italic and bold.
- The `REGULAR` keyword removes any special style characteristics. When used with other style keywords, it must be specified first; otherwise any previous style specifications are removed. For example, `STYLE=REGULAR BOLD` results in bold text without any other special style characteristics, but `STYLE=BOLD REGULAR` results in text that is not bold.

TEXTCOLOR Keyword

The `TEXTCOLOR` keyword specifies the text color.

- The keyword is followed by an equals sign and a color keyword or `RGB(r, g, b)`, where r, g, and b are numeric red, green, and blue color values. The numeric values are integers between 1 and 255.
- Valid color keywords are `BLACK, BLUE, CYAN, DARKGRAY, GRAY, GREEN, LIGHTGRAY, MAGENTA, ORANGE, PINK, RED, WHITE, YELLOW`.

REPORT Subcommand

The `REPORT` subcommand produces a table that summarizes the results of the `OUTPUT MODIFY` command. If included, this subcommand must be the first subcommand.

PRINTREPORT=NO
Do not include a summary report table. This option is the default.

PRINTREPORT=YES
Include a summary report table.

POWER ONEWAY ANOVA

POWER ONEWAY ANOVA is available in IBM SPSS Statistics Base Edition. The procedure estimates the power for two types of hypothesis to compare the multiple group means, the overall test, and the test with specified contrasts. The over test focuses on the null hypothesis that all group means are equal. The test with specified contrasts breaks down the overall ANOVA hypotheses into smaller but more describable and useful pieces of the means.

```
POWER ONEWAY ANOVA
     /PARAMETERS
          [ SIGNIFICANCE={0.05**} {value}] ]
          { GROUP_SIZES={integers} } {POWER={value} }
          POOLED_SD={value}
          GROUP_MEANS={values}
          [ GROUP_WEIGHTS={EQUAL**} {values} ]
     [ /CONTRAST
          [ COEFFICIENTS ={values}]
          [ TEST ={NONDIRECTIONAL**} {DIRECTIONAL}]
          [ PAIRWISE={FALSE**} {TRUE}]
          [ ADJUST={BONFERRONI**} {SIDAK} {LSD}] ]
     [ /PLOT
          [ TOTAL_N([integer1 integer2]) ]
          [ POOLED_SD([value1 value2]) ]
          [ TOTAL_N_XAXIS([integer1 integer2]) POOLED_SD_YAXIS([value1 value2]) ]
          [ TOTAL_N_YAXIS([integer1 integer2]) POOLED_SD_XAXIS([value1 value2]) ]
          [ VROTATE({10**} {integer}) ]
          [ HROTATE({325**} {integer}) ] ]
```

** Default if the subcommand or keyword is omitted.

This command reads the active dataset and causes execution of any pending commands. See the topic Command Order for more information.

Release History

Release 27.0

- Command introduced

Overview

`POWER ONEWAY ANOVA` is an extension of the two sample *t*-test procedure, and focuses on the one-way ANOVA and statistical inference about population means for multiple groups. The procedure estimates the power for two types of hypothesis to compare the multiple group means, the overall test, and the test with specified contrasts. The over test focuses on the null hypothesis that all group means are equal. The test with specified contrasts breaks down the overall ANOVA hypotheses into smaller but more describable and useful pieces of the means.

Options

Test Assumptions
Specifies the keywords that control the test assumptions. The keywords specify the group sizes, the pooled population standard deviation, and the group means.

Significance Level
Specifies the significance level of the test's Type I error rate.

Contrast Test
Specifies the contrast coefficients and requests the contrast test and determines whether the contrast test is one or two sided.

Pairwise Differences
Controls whether or not to estimate the power of testing for the pairwise differences, and determines the adjustment of multiple comparisons.

Two-Dimensional Plot
Controls the two-dimensional power by total sample size chart, and the two-dimensional power by pooled standard deviation chart.

Three-Dimensional Plot

Controls the three-dimensional power by total sample size (x-axis) and pooled standard deviation (y-axis) chart, and the three-dimensional power by total sample size (y-axis) and pooled standard deviation (x-axis) chart.

Basic specification

- `POWER ONEWAY ANOVA`: Invokes the Power Analysis of One-way ANOVA with Contrasts procedure.

Subcommand order

- The subcommands can be named in any order.

Example

```
POWER ONEWAY ANOVA
  /PARAMETERS GROUP_SIZES=2 2 POOLED_SD=1 GROUP_MEANS=2 2 SIGNIFICANCE=0.05
  /CONTRAST PAIRWISE=TRUE ADJUST=BONFERRONI
  /PLOT TOTAL_N TOTAL_N_YAXIS POOLED_SD_XAXIS VROTATE(10) HROTATE(325).
```

- **PARAMETERS** specifies that both group sizes are set to 2, the pooled population standard deviation is set to 1, both group means are set to 2, and the significance level uses the default 0.05 value.
- **CONTRAST** specifies that estimating the power of testing for the pairwise differences is enabled and the adjustment of multiple comparisons uses the Bonferroni correction in estimating the power of pairwise differences.
- **PLOT** specifies that the two-dimensional power by total sample size chart displays in the output. The three-dimensional power by total sample size (x -axis) and pooled standard deviation (y-axis) chart also displays in the output. The vertical and horizontal rotation settings use their default values (10 degrees and 325 degrees respectively).

PARAMETERS Subcommand

PARAMETERS controls the test assumptions. The keywords specify the group sizes, the pooled population standard deviation, and the group means.

SIGNIFICANCE = value

An optional keyword that specifies the significance level of the Type I error rate for the test. The value must be a single double value ∈`(0,1)`. The default level is `0.05`.

GROUP_SIZES = integers

A required keyword that specifies the group sizes. At least two integers are required. Each integer must be ≥ 2. Multiple integers are separated by a blank space (or blank spaces). The number of specified integers must be equal to that for **GROUP_MEANS**.

POWER=value

A required keyword that specifies the power for sample size estimation. The value must be a single value ∈`(0,1)`. **POWER** and **GROUP_SIZES** cannot be specified simultaneously.

POOLED_SD = value

A required keyword that specifies the pooled population standard deviation. The value must be a single, numeric value > 0.

GROUP_MEANS = values

A required keyword that specifies the group means. At least two values are required. Multiple values are separated by a blank space (or blank spaces). The number of specified values must be equal to that for **GROUP_SIZES**.

GROUP_WEIGHTS = values

A required keyword that assigns the group size weights when **POWER** is specified. The keyword takes effect when **POWER** is specified, and is ignored when **GROUP_SIZES** is specified. The default setting is **EQUAL**, which assigns the group weights all equal 1. The keyword can also be reset by specifying multiple values, separated by a blank space (or blank spaces). Each value must be ∈`[1,100]`. The number of specified values must be equal to that for **GROUP_MEANS**.

CONTRAST Subcommand

The optional `CONTRAST` subcommand specifies the contrasts and pairwise differences.

COEFFICIENTS = values
The optional keyword specifies the contrast coefficients and requests the contrast test. The keyword is omitted by default. When the keyword is specified, the number of specified values must be equal to that for **/PARAMETERS GROUP_SIZES** or **/PARAMETERS GROUP_MEANS**. All of the specified values must be summed up to `0`, otherwise the last value is automatically adjusted.

TEST = {NONDIRECTIONAL}{DIRECTIONAL}
The optional keyword determines whether the contrast test is one or two-sided. **NONDIRECTIONAL**, or a two-sided test, is the default setting. When **DIRECTIONAL** is specified, power is computed for a one-sided test. The keyword takes effect when **/CONTRAST COEFFICIENTS** is specified.

PAIRWISE = boolean
The optional keyword controls whether or not to estimate the power of testing for the pairwise differences. **FALSE** is the default setting, which suppresses output for the pairwise differences. When **TRUE** is specified Table 3 is required and displays. In cases where **COEFFICIENTS** is omitted and **PAIRWISE = FALSE**, a warning message is created that indicates:

```
COEFFICIENTS is required
```

ADJUST = {BONFERRONI} {SIDAK} {LSD}
The optional keyword determines the adjustment of multiple comparisons. **BONFERRONI** is the default setting, which uses the Bonferroni correction in estimating the power of pairwise differences. **SIDAK** and **LSD** use the Sidak correction and the least significant difference, respectively. The keyword takes effect when **/CONTRAST PAIRWISE = TRUE** is specified.

PLOT Subcommand

The optional `PLOT` subcommand controls the output charts. By default, no charts are created. When the subcommand is specified, only the outputs for the following keywords are created.

TOTAL_N(integer1 integer2)
The optional keyword controls the two-dimensional power by total sample size chart. By default, the chart is suppressed. If specified, the chart will be displayed. When no integers are specified, the default plot range of the sample size ratio is used. When the keyword is specified, the plot boundaries are determined by **integer1** and **integer2**, respectively. Both **integer1** and **integer2** are integers, separated by a blank space (or blank spaces), and must satisfy `integer1 ≥ 2*` the number of integers specified for **/PARAMETERS GROUP_SIZES**, `integer1 < integer2`, and `integer2 ≤ 5000`. You can either omit or specify both numeric values.

POOLED_SD(value1 value2)
The optional keyword controls the two-dimensional power by pooled standard deviation chart. The chart is suppressed by default. When the keyword is specified, the chart displays. When no values are specified, the default plot range of the pooled standard deviation is used. When values are specified, the plot boundaries are determined by **value1** and **value2**, respectively. Both **value1** and **value2** are numeric values, separated by a blank space (or blank spaces) and must satisfy `0 < value1 < value2`. You can either omit or specify both numeric values.

TOTAL_N_XAXIS(integer1 integer2) POOLED_SD_YAXIS(value1 value2)
The optional keyword controls the three-dimensional power by total sample size (x-axis) and pooled standard deviation (y-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

Note: **integer1**, **integer2**, **value1** and **value2** follow the same rules as specified for **TOTAL_N** and **POOLED_SD**.

TOTAL_N_YAXIS(integer1 integer2) POOLED_SD_XAXIS(value1 value2)
The optional keyword controls the three-dimensional power by total sample size (y-axis) and pooled standard deviation (x-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

Note: **`integer1`**, **`integer2`**, **`value1`** and **`value2`** follow the same rules as specified for **`TOTAL_N`** and **`POOLED_SD`**.

VROTATE(integer)
The optional keyword sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer ∊`[0,359]` with `10` as the default setting.

HROTATE(integer)
The optional keyword sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer ∊`[0,359]` with 325 as the default setting.

POWER MEANS INDEPENDENT

POWER MEANS INDEPENDENT is available in IBM SPSS Statistics Base Edition. The command invokes power analysis for the independent sample *t*-test to draw statistical inference about the difference of the two means. It is assumed that the data in each sample independently and identically follow a normal distribution with a fixed mean and variance.

```
POWER MEANS INDEPENDENT
     /PARAMETERS
          [ TEST={NONDIRECTIONAL**} {DIRECTIONAL}]
          [ SIGNIFICANCE={0.05**} {value}] ]
          { N={integer1 integer2} {POWER={value} }
          [ NRATIO={1**} {value} ]
          SD={value} {value1 value2}
          MEAN={value} {value1 value2}
     [ /PLOT
          [ NRATIO([value1 value2]) ]
          [ ES([value1 value2]) ]
          [ NRATIO_XAXIS([integer1 integer2]) ES_YAXIS([value1 value2]) ]
          [ NRATIO_YAXIS([integer1 integer2]) ES_XAXIS([value1 value2]) ]
          [ VROTATE({10**} {integer}) ]
          [ HROTATE({325**} {integer}) ] ]
```

** Default if the subcommand or keyword is omitted.

This command reads the active dataset and causes execution of any pending commands. See the topic Command Order for more information.

Release History

Release 27.0

- Command introduced

Overview

`POWER MEANS INDEPENDENT` invokes power means analysis for the independent sample *t*-test.

Options

Test Assumptions
Specifies the keywords that control the test assumptions. The keywords specify the sample size, population standard deviation, and the population mean under testing. For more information, see “PARAMETERS Subcommand ” on page 145.

Significance Level
Specifies the significance level of the test's Type I error rate.

Test Direction
Determines whether the test is one or two-sided.

Two-Dimensional Plot
Controls the two-dimensional power by sample size ratio chart, and the two-dimensional power by effect size (or mean difference) chart.

Three-Dimensional Plot

Controls the three-dimensional power by sample size (x-axis) and effect size (y-axis) chart, and the three-dimensional power by sample size (y-axis) and effect size (x-axis) chart. Also controls the vertical rotation degrees (clockwise from the left) and the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart.

Basic specification

- `POWER MEANS INDEPENDENT`: Invokes the power analysis for the independent sample *t*-test.

Subcommand order

- The subcommands can be named in any order.

Example

```
POWER MEANS INDEPENDENT
  /PARAMETERS TEST=NONDIRECTIONAL SIGNIFICANCE=0.05 N=2 4 MEAN=1 SD=.5
  /PLOT ES NRATIO_YAXIS ES_XAXIS VROTATE(10) HROTATE(325).
```

- **PARAMETERS** specifies that the test is non-directional (two-sided), the significance level uses the default 0.05 value, the population mean(s) under testing is set to 1, and the population standard deviation is set to 0.5.
- **PLOT** specifies that the two-dimensional power by effect size (or mean difference) chart displays in the output. The three-dimensional power by sample size ratio (y-axis) and effect size (x -axis) chart also displays in the output. The vertical and horizontal rotation settings use their default values (10 degrees and 325 degrees respectively).

PARAMETERS Subcommand

PARAMETERS controls the test assumptions. The keywords specify the sample sizes, population standard deviations, and the population means under testing.

TEST = {NONDIRECTIONAL}{DIRECTIONAL}

An optional keyword that determines whether the test is one or two-sided. **NONDIRECTIONAL**, or a two-sided test, is the default setting. When **DIRECTIONAL** is specified, power is computed for a one-sided test.

SIGNIFICANCE = value

An optional keyword that specifies the significance level of the Type I error rate for the test. The value must be a single double value $\in (0,1)$. The default level is `0.05`.

N = {integer1 integer2}

A required keyword that specifies the sample size for the two groups for comparison. The values for both integers must be > 1.

POWER = value

A required keyword that specifies the power for sample size estimation. The value must be a single value $\in (0,1)$. **POWER** and **N** cannot be specified simultaneously.

NRATIO = {value}

A required keyword that specifies the ratio of the sample sizes. Must be a single value $\in [0.01,100]$ with 1 as the default setting. The keyword takes effect when **POWER** is specified, and is ignored when **N** is specified.

SD = {value}{value1 value2}

A required keyword that specifies the population standard deviation(s). Note that **value**, **value1**, and **value2** are all numeric values > 0. Multiple parameters **value1** and **value2**, if specified, are separated by a blank space (or blank spaces). When a single value is specified, it denotes the population standard deviation of the group difference σ, and assumes that the two group variances are equal, or $\sigma_1 = \sigma_2 = \sigma$. When two different values (**value1** and **value2**) are specified, they denote σ_1 and σ_2, respectively.

Note: When two identical **`value1`** and **`value2`** values are specified, the result is the same as a when a single value is specified for the keyword.

MEAN = {value}{value1 value2}

A required keyword that specifies the population mean(s) under testing. Note that **`value`**, **`value1`**, and **`value2`** are all numeric values. Multiple parameters **`value1`** and **`value2`**, if specified, are separated by a blank space (or blank spaces). When a single value is specified, it denotes the population mean difference μ_d. When two values (**`value1`** and **`value2`**) are specified, they denote μ_1 and μ_2, respectively.

PLOT Subcommand

The optional PLOT subcommand controls the output charts. By default, no charts are created. When the subcommand is specified, only the outputs for the following keywords are created.

NRATIO(value1 value2)

The optional keyword controls the two-dimensional power by sample size ratio chart. By default, the chart is suppressed. If specified, the chart will be displayed. When no integers are specified, the default plot range of the sample size ratio is used (depending on whether the group variances are assumed to be equal or not). When the keyword is specified, the plot boundaries are determined by **`value1`** and **`value2`**, respectively. Both **`value1`** and **`value2`** are numeric values, separated by a blank space (or blank spaces), and must satisfy `0.2 ≤ value1 < value2 ≤ 5.0`. You can either omit or specify both numeric values.

ES(value1 value2)

The optional keyword controls the two-dimensional power by effect size (or mean difference) chart. The chart is suppressed by default. When the keyword is specified, the chart displays. When no values are specified, the default plot range of the effect size (or mean difference) is used, depending wither the group variances are assumed to be equal or not. When values are specified, the plot boundaries are determined by **`value1`** and **`value2`**, respectively. Both **`value1`** and **`value2`** are numeric values, separated by a blank space (or blank spaces). For the equal group variances, it must satisfy `−5.0 ≤ value1 < value2 ≤ 5.0` when **`/CRITERIA TEST = DIRECTIONAL`**, or `0 ≤ value1 < value2 ≤ 5.0` when **`/CRITERIA TEST = NONDIRECTIONAL`**. For the unequal group variances, it must satisfy `0 ≤ value1 < value2 ≤ 5000`. You can either omit or specify both numeric values.

NRATIO_XAXIS(integer1 integer2) ES_YAXIS(value1 value2)

The optional keyword controls the three-dimensional power by sample size (x-axis) and effect size (y-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

Note: `value1` and **`value2`** follow the same rules as specified for **`NRATIO`** and **`ES`**.

NRATIO_YAXIS(integer1 integer2) ES_XAXIS(value1 value2)

The optional keyword controls the three-dimensional power by sample size (y-axis) and effect size (x-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

Note: `value1` and **`value2`** follow the same rules as specified for **`NRATIO`** and **`ES`**.

VROTATE(integer)

The optional keyword sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer ∊`[0,359]` with `10` as the default setting.

HROTATE(integer)

The optional keyword sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer ∊`[0,359]` with `325` as the default setting.

POWER MEANS ONESAMPLE

POWER MEANS ONESAMPLE is available in IBM SPSS Statistics Base Edition. The command invokes power analysis for the one sample *t*-test to draw statistical inference about the mean parameter. It is assumed that the sample data independently and identically follow a normal distribution with a fixed mean and variance.

```
POWER MEANS ONESAMPLE
     /PARAMETERS
          [ TEST={NONDIRECTIONAL**} {DIRECTIONAL} ]
          [ SIGNIFICANCE={0.05**} {value}] ]
          { N={integer} {POWER={value} }
          SD={value}
          MEAN={value}
          [ NULL={0**} {value} ]
     [ /PLOT
          [ N([integer1 integer2]) ]
          [ ES([value1 value2]) ]
          [ N_XAXIS([integer1 integer2]) ES_YAXIS([value1 value2]) ]
          [ N_YAXIS([integer1 integer2]) ES_XAXIS([value1 value2]) ]
          [ VROTATE({10**} {integer}) ]
          [ HROTATE({325**} {integer}) ] ]
```

** Default if the subcommand or keyword is omitted.

This command reads the active dataset and causes execution of any pending commands. See the topic Command Order for more information.

Release History

Release 27.0

- Command introduced

Overview

`POWER MEANS ONESAMPLE` invokes the power means analysis for the one sample *t*-test.

Options

Test Assumptions
Specifies the keywords that control the test assumptions. The keywords specify the sample size, population standard deviation, population mean under testing, and hypothesized value to be tested. For more information, see "PARAMETERS Subcommand " on page 148.

Significance Level
Specifies the significance level of the test's Type I error rate.

Test Direction
Determines whether the test is one or two-sided.

Two-Dimensional Plot
Controls the two-dimensional power by sample size chart, and the two-dimensional power by effect size chart.

Three-Dimensional Plot
Controls the three-dimensional power by sample size (x-axis) and effect size (y-axis) chart, and the three-dimensional power by sample size (y-axis) and effect size (x-axis) chart. Also controls the vertical rotation degrees (clockwise from the left) and the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart.

Basic specification

- `POWER MEANS ONESAMPLE`: Invokes the power analysis for the one sample *t*-test.

Subcommand order

- The subcommands can be named in any order.

Example

```
POWER MEANS ONESAMPLE
  /PARAMETERS TEST=NONDIRECTIONAL SIGNIFICANCE=0.05 N=2 SD=1 MEAN=1 NULL=.5
  /PLOT N(2 5) N_YAXIS(2 5) ES_XAXIS VROTATE(10) HROTATE(325).
```

- **PARAMETERS** specifies that the test is non-directional (two-sided), the significance level uses the default 0.05 value, the sample size for power estimation is set to 1, the population standard deviation is set to 0.5, the population mean(s) under testing is set to 1, and the null hypothesis value to be tested is set to 0.5.
- **PLOT** specifies that the two-dimensional power by sample size chart displays in the output. The three-dimensional power by sample size (x -axis) and effect size (y-axis) chart also displays in the output. The vertical and horizontal rotation settings use their default values (10 degrees and 325 degrees respectively).

PARAMETERS Subcommand

PARAMETERS controls the test assumptions. The keywords specify the sample size, population standard deviation, population mean under testing, and hypothesized value to be tested.

TEST = {NONDIRECTIONAL}{DIRECTIONAL}
An optional keyword that determines whether the test is one or two-sided. **NONDIRECTIONAL**, or a two-sided test, is the default setting. When **DIRECTIONAL** is specified, power is computed for a one-sided test.

SIGNIFICANCE = value
An optional keyword that specifies the significance level of the Type I error rate for the test. The value must be a single double value `∊(0,1)`. The default level is `0.05`.

N = integer
A required keyword that specifies the sample size. The value must be a single integer > 1.

POWER = value
A required keyword that specifies the power for sample size estimation. The value must be a single value `∊(0,1)`. **POWER** and **N** cannot be specified simultaneously.

SD = value
A required keyword that specifies the population standard deviation. The value must be a single, numeric value > 0.

MEAN = value
A required keyword that specifies the population mean under testing. The value must be a single, numeric value.

NULL = value
An optional keyword that specifies the null hypothesis value to be tested. Must be a single numeric value. The default setting is 0.

PLOT Subcommand

The optional PLOT subcommand controls the output charts. By default, no charts are created. When the subcommand is specified, only the outputs for the following keywords are created.

N(integer1 integer2)
The optional keyword controls the two-dimensional power by sample size chart. The chart is suppressed by default. When specified, the chart displays and the plot boundaries are determined by **integer1** and **integer2**, respectively. When no integers are specified, the default plot range is used. Both **integer1** and **integer2** are integers, separated by a blank space (or blank spaces), and must satisfy: `1 < integer1 < integer2 ≤ 5000`. You can either omit or specify both integer values.

ES(value1 value2)
The optional keyword controls the two-dimensional power by effect size chart. The chart is suppressed by default. When the keyword is specified, the chart displays. When no values are specified, the default plot range of the effect size is used. When values are specified, the plot boundaries are determined by **value1** and **value2**, respectively. Both **value1** and **value2** are numeric values, separated by a blank space (or blank spaces), and must satisfy `–5.0 ≤ value1 < value2 ≤ 5.0` when **/CRITERIA TEST = DIRECTIONAL**, or `0 ≤ value1 < value2 ≤ 5.0` when **/CRITERIA TEST = NONDIRECTIONAL**. You can either omit or specify both numeric values.

N_XAXIS(integer1 integer2) ES_YAXIS(value1 value2)

The optional keyword controls the three-dimensional power by sample size (x-axis) and effect size (y-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

Note: **`integer1`**, **`integer2`**, **`value1`**, and **`value2`** follow the same rules as specified for **`N`** and **`ES`**.

N_YAXIS(integer1 integer2) ES_XAXIS(value1 value2)

The optional keyword controls the three-dimensional power by sample size (y-axis) and effect size (x-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

Note: **`integer1`**, **`integer2`**, **`value1`**, and **`value2`** follow the same rules as specified for **`N`** and **`ES`**.

VROTATE(integer)

The optional keyword sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer ∈`[0,359]` with `10` as the default setting.

HROTATE(integer)

The optional keyword sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer ∈`[0,359]` with 325 as the default setting.

POWER MEANS RELATED

POWER MEANS RELATED is available in IBM SPSS Statistics Base Edition. The command invokes power analysis for the related sample *t*-test to draw statistical inference about the difference of the two means. It is assumed that the data in each sample independently and identically follow a normal distribution with a fixed mean and variance.

```
POWER MEANS RELATED
     /PARAMETERS
          [ TEST={NONDIRECTIONAL**} {DIRECTIONAL} ]
          [ SIGNIFICANCE={0.05**} {value}] ]
          { NPAIRS={integer} {POWER={value} }
          SD={value} {value1 value2}
          [ RHO ={0.5**} {value}]
          MEAN={value} {value1 value2}
     [ /PLOT
          [ NPAIRS([integer1 integer2]) ]
          [ ES([value1 value2]) ]
          [ NPAIRS_XAXIS([integer1 integer2]) ES_YAXIS([value1 value2]) ]
          [ NPAIRS_YAXIS([integer1 integer2]) ES_XAXIS([value1 value2]) ]
          [ VROTATE({10**} {integer}) ]
          [ HROTATE({325**} {integer}) ] ]
```

** Default if the subcommand or keyword is omitted.

This command reads the active dataset and causes execution of any pending commands. See the topic Command Order for more information.

Release History

Release 27.0

- Command introduced

Overview

`POWER MEANS RELATED` invokes the power means analysis for the related samples *t*-test.

Options

Test Assumptions

Specifies the keywords that control the test assumptions. The keywords specify the number of group pairs, the population standard deviation(s), the Pearson product-moment correlation co-efficient ρ, and the population mean(s) under testing. For more information, see "PARAMETERS Subcommand " on page 150.

Significance Level

Specifies the significance level of the test's Type I error rate.

Test Direction
Determines whether the test is one or two-sided.

Two-Dimensional Plot
Controls the two-dimensional power by sample size chart, and the two-dimensional power by effect size chart.

Three-Dimensional Plot
Controls the three-dimensional power by sample size (x-axis) and effect size (y-axis) chart, and the three-dimensional power by sample size (y-axis) and effect size (x-axis) chart. Also controls the vertical rotation degrees (clockwise from the left) and the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart.

Basic specification

- `POWER MEANS RELATED`: Invokes the power analysis for the related samples *t*-test.

Subcommand order

- The subcommands can be named in any order.

Example

```
POWER MEANS RELATED
  /PARAMETERS TEST=NONDIRECTIONAL SIGNIFICANCE=0.05 NPAIRS=4 SD=.75 MEAN=1
  /PLOT NPAIRS(2 4) NPAIRS_YAXIS(2 4) ES_XAXIS VROTATE(10) HROTATE(325).
```

- **PARAMETERS** specifies that the test is non-directional (two-sided), the significance level uses the default 0.05 value, the number of group pairs for power estimation is set to 4, the population standard deviation is set to 0.75, and the population mean under testing is set to 1.
- **PLOT** specifies that the two-dimensional power by sample size chart displays in the output. The three-dimensional power by sample size (x -axis) and effect size (y-axis) chart also displays in the output. The vertical and horizontal rotation settings use their default values (10 degrees and 325 degrees respectively).

PARAMETERS Subcommand

PARAMETERS controls the test assumptions. The keywords specify the number of group pairs, the population standard deviation(s), the Pearson product-moment correlation co-efficient ρ, and the population mean(s) under testing.

TEST = {NONDIRECTIONAL}{DIRECTIONAL}
An optional keyword that determines whether the test is one or two-sided. **NONDIRECTIONAL**, or a two-sided test, is the default setting. When **DIRECTIONAL** is specified, power is computed for a one-sided test.

SIGNIFICANCE = value
An optional keyword that specifies the significance level of the Type I error rate for the test. The value must be a single double value $\in$`(0,1)`. The default level is `0.05`.

NPAIRS = integer
A required keyword that specifies the number of group pairs. The value must be a single integer > 1. **NPAIRS** and **POWER** cannot be specified simultaneously.

POWER=value
A required keyword that specifies the power for sample size estimation. The value must be a single value $\in$`(0,1)`. POWER and NPAIRS cannot be specified simultaneously.

SD = {value} {value1 value2}
A required keyword that specifies the population standard deviation(s). Note that **value**, **value1**, and **value2** are all numeric values > 0. Multiple parameters **value1** and **value2**, if specified, are separated by a blank space (or blank spaces). When a single value is specified, it denotes the population standard deviation of the group difference σ_d. When two values (**value1** and **value2**) are specified, they denote σ_1 and σ_2, respectively.

RHO = value
An optional keyword that specifies the Pearson product-moment correlation coefficient ρ. The value must be a single numeric $\in$`[-1,1]` and cannot be `0`. The default value is `0.5`. Note that this keyword is ignored when a single value is specified for **SD**. Otherwise, when two values (**value1** and **value2**) are specified for **SD**, this keyword takes effect and is used to compute σ_d. In cases where **value1=value2** is specified for **SD**, ρ=1 is not allowed.

MEAN = {value} {value1 value2}
A required keyword that specifies the population mean(s) under testing. Note that **value**, **value1**, and **value2** are all numeric values. Multiple parameters **value1** and **value2**, if specified, are separated by a blank space (or blank spaces). When a single value is specified, it denotes the population mean difference μ_d. When two values(**value1** and **value2**) are specified, they denote μ_1 and μ_2, respectively.

PLOT Subcommand

The optional PLOT subcommand controls the output charts. By default, no charts are created. When the subcommand is specified, only the outputs for the following keywords are created.

NPAIRS(integer1 integer2)
The optional keyword controls the two-dimensional power by sample size chart. The chart is suppressed by default. When specified, the chart displays and the plot boundaries are determined by **integer1** and **integer2**, respectively. When no integers are specified, the default plot range is used. Both **integer1** and **integer2** are integers, separated by a blank space (or blank spaces), and must satisfy: `1 < integer1 < integer2 ≤ 5000`. You can either omit or specify both integer values.

ES(value1 value2)
The optional keyword controls the two-dimensional power by effect size chart. The chart is suppressed by default. When the keyword is specified, the chart displays. When no values are specified, the default plot range of the effect size is used. When values are specified, the plot boundaries are determined by **value1** and **value2**, respectively. Both **value1** and **value2** are numeric values, separated by a blank space (or blank spaces), and must satisfy `−5.0 ≤ value1 < value2 ≤ 5.0` when **/CRITERIA TEST = DIRECTIONAL**, or `0 ≤ value1 < value2 ≤ 5.0` when **/CRITERIA TEST = NONDIRECTIONAL**. You can either omit or specify both numeric values.

NPAIRS_XAXIS(integer1 integer2) ES_YAXIS(value1 value2)
The optional keyword controls the three-dimensional power by sample size (x-axis) and effect size (y-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

Note: **integer1**, **integer2**, **value1**, and **value2** follow the same rules as specified for **NPAIRS** and **ES**.

NPAIRS_YAXIS(integer1 integer2) ES_XAXIS(value1 value2)
The optional keyword controls the three-dimensional power by sample size (y-axis) and effect size (x-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

Note: **integer1**, **integer2**, **value1**, and **value2** follow the same rules as specified for **NPAIRS** and **ES**.

VROTATE(integer)
The optional keyword sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer $\in$`[0,359]` with `10` as the default setting.

HROTATE(integer)
The optional keyword sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer $\in$`[0,359]` with 325 as the default setting.

POWER PARTIALCORR

`POWER PARTIALCORR` is available in IBM SPSS Statistics Base Edition. The command invokes the power analysis for the one-sample partial correlation test. Partial correlation can be explained as the association between two random variables after eliminating the effect of another or several other variables. It is a useful measurement in the presence of confounding

```
POWER PARTIALCORR
    /PARAMETERS
        [ TEST = {NONDIRECTIONAL**} {DIRECTIONAL} ]
        [ SIGNIFICANCE = {0.05**} {value} ]
        PARTIALOUT = {integer}
        { N = {integer} } { POWER = {value} }
        [ NULL = {0**} {value} ]
        ALTERNATIVE = {value}
    [ /PLOT
        [ N([integer1 integer2 ]) ] [ PARTIALOUT ] [ NULL ] [ ALTERNATIVE ] [ DIFF ]
        [ N_XAXIS([integer1 integer2]) DIFF_YAXIS ]
        [ N_YAXIS([integer1 integer2]) DIFF_XAXIS ]
        [ NULL_XAXIS ALTERNATIVE_YAXIS ]
        [ NULL_YAXIS ALTERNATIVE_XAXIS ]
        [ VROTATE({10**} {integer}) ]
        [ HROTATE({325**} {integer}) ]
```

** Default if the subcommand or keyword is omitted.

This command reads the active dataset and causes execution of any pending commands. See the topic Command Order for more information.

Release History

Release 27.0

- Command introduced

Overview

`POWER PARTIALCORR` invokes the power analysis for the one-sample partial correlation test.

Options

Test Assumptions
Specifies the keywords that control the test assumptions. The keywords specify the sample size, power for sample size estimation, null hypothesis value of the partial correlation parameter, alternative hypothesis value of the partial correlation parameter, test direction, variables assumed to be partialled-out, and the significance level of the type I error rate. For more information, see "PARAMETERS Subcommand " on page 153.

Two-Dimensional Plot
Controls the two-dimensional power by number of partialled-out variables chart, the two-dimensional power by null value chart, the two-dimensional power by alternative value chart, and the two-dimensional power by difference between hypothesized values chart.

Three-Dimensional Plot
Controls the three-dimensional power by sample size (x-axis) and difference between hypothesized values (y-axis) chart, the three-dimensional power by sample size (y-axis) and difference between hypothesized values (x-axis) chart, the three-dimensional power by null (x-axis) and alternative value (y-axis) chart, and the three-dimensional power by null (y-axis) and alternative value (x-axis) chart. Also controls the vertical rotation degrees (clockwise from the left) and the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart.

Basic specification

- `POWER PARTIALCORR`: Invokes the power analysis for the one-sample partial correlation test.

Subcommand order

- The subcommands can be named in any order.

Example

```
POWER PARTIALCORR
  /PARAMETERS TEST=NONDIRECTIONAL SIGNIFICANCE=0.05
      PARTIALOUT=1 N=5 NULL=0 ALTERNATIVE=1
  /PLOT NULL NULL_XAXIS ALTERNATIVE_YAXIS VROTATE(10)
      HROTATE(325).
```

- **PARAMETERS** specifies that the test direction is non-directional, the significance level uses the default 0.05 value, the variables assumed to be partialled-out is set to 1, the sample size is set to 5, the null hypothesis value of the partial correlation parameter is set to 0, and the alternative hypothesis value of the partial correlation parameter is set to 1.
- **PLOT** specifies that the two-dimensional power by null value chart displays in the output. The three-dimensional power by null (x-axis) and alternative value (y-axis) chart also displays in the output. The vertical and horizontal rotation settings use their default values (10 degrees and 325 degrees respectively).

PARAMETERS Subcommand

PARAMETERS controls the test assumptions. The keywords specify the sample size, power for sample size estimation, null hypothesis value of the partial correlation parameter, alternative hypothesis value of the partial correlation parameter, test direction, variables assumed to be partialled-out, and the significance level of the type I error rate.

TEST = {NONDIRECTIONAL}{DIRECTIONAL}

An optional keyword that determines whether the test is one or two-sided. **NONDIRECTIONAL**, or a two-sided test, is the default setting. When **DIRECTIONAL** is specified, power is computed for a one-sided test.

SIGNIFICANCE = value

An optional keyword that specifies the significance level of the Type I error rate for the test. The value must be a single double value ∈`(0,1)`. The default level is `0.05`.

PARTIALOUT = integer

A required keyword to specify the number of the variables assumed to be partialled-out. Must be a single integer ≥ 0.

N = integer

A required keyword that specifies the sample size. The value must be a single integer > **PARTIALOUT** + 3. **N** and **POWER** cannot be specified simultaneously.

POWER = value

A required keyword that specifies the power for sample size estimation. The value must be a single value ∈`(0,1)`. **POWER** and **N** cannot be specified simultaneously.

NULL = value

An optional keyword that specifies the null hypothesis value of the partial correlation parameter. Must be a single numeric value ∈`(-1,1)`. The default setting is 0. When **POWER** is specified, the **NULL** value cannot be -1 or 1.

ALTERNATIVE = value

A required keyword that specifies the alternative hypothesis value of the partial correlation parameter. Must be a single numeric value ∈`(-1,1)`. When **POWER** is specified, the **ALTERNATIVE** value cannot be -1 or 1, or equal to the **NULL** value.

PLOT Subcommand

The optional PLOT subcommand controls the output charts. By default, no charts are created. When the subcommand is specified, only the outputs for the following keywords are created.

N(integer1 integer2)

The optional keyword controls the two-dimensional power by sample size chart. The chart is suppressed by default. When specified, the chart displays and the plot boundaries are determined by **integer1** and **integer2**, respectively. When no integers are specified, the default plot range is used. Both **integer1** and **integer2** are integers, separated by a blank space (or blank spaces), and must satisfy: **`PARTIALOUT`** `+ 4 ≤ integer1 < integer2 ≤` **`PARTIALOUT`** `+ 5000`. You can either omit or specify both integer values.

PARTIALOUT

The optional keyword controls the two-dimensional power by number of partialled-out variables chart. The chart is suppressed by default. When specified, the chart displays.

NULL

The optional keyword controls the two-dimensional power by null value chart. The chart is suppressed by default. When specified, the chart displays.

ALTERNATIVE

The optional keyword controls the two-dimensional power by alternative value chart. The chart is suppressed by default. When specified, the chart displays.

DIFF

The optional keyword controls the two-dimensional power by risk difference chart. The chart is suppressed by default. When specified, the chart displays.

N_XAXIS(integer1 integer2) DIFF_YAXIS

The optional keyword controls the three-dimensional power by total number of trials (x -axis) and risk difference (y-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

Note: **`integer1`**and **`integer2`** follow the same rules as specified for **`N`**.

N_YAXIS(integer1 integer2) DIFF_XAXIS

The optional keyword controls the three-dimensional power by total number of trials (y-axis) and risk difference (x -axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

Note: **`integer1`**and **`integer2`** follow the same rules as specified for **`N`**.

NULL_XAXIS ALTERNATIVE_YAXIS

The optional keyword controls the three-dimensional power by null (x -axis) and alternative value (y-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

NULL_YAXIS ALTERNATIVE_XAXIS

The optional keyword controls the three-dimensional power by null (y-axis) and alternative value (x -axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

VROTATE(integer)

The optional keyword sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer ∈`[0,359]` with `10` as the default setting.

HROTATE(integer)

The optional keyword sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer ∈`[0,359]` with 325 as the default setting.

POWER PEARSON ONESAMPLE

`POWER PEARSON ONESAMPLE` is available in IBM SPSS Statistics Base Edition. The command invokes the power analysis for the one-sample Pearson correlation test. Pearson product-moment correlation coefficient measures the strength of linear association between two scale random variables, which are assumed to follow a bivariate normal distribution.

```
POWER PEARSON ONESAMPLE
    /PARAMETERS
        [ TEST = {NONDIRECTIONAL**} {DIRECTIONAL} ]
        [ ADJUST BIAS = {TRUE**} {FALSE} ]
        [ SIGNIFICANCE = {0.05**} {value} ]
        { N = {integer} } { POWER = {value} }
        [ NULL = {0**} {value} ]
        ALTERNATIVE = {value}
    [ /PLOT
        [ N([integer1 integer2 ]) ] [ NULL ] [ ALTERNATIVE ] [ DIFF ]
        [ N_XAXIS([integer1 integer2]) DIFF_YAXIS ]
```

```
        [ N_YAXIS([integer1 integer2]) DIFF_XAXIS ]
        [ NULL_XAXIS ALTERNATIVE_YAXIS ]
        [ NULL_YAXIS ALTERNATIVE_XAXIS ]
        [ VROTATE({10**} {integer}) ]
        [ HROTATE({325**} {integer}) ] ]
```

** Default if the subcommand or keyword is omitted.

This command reads the active dataset and causes execution of any pending commands. See the topic Command Order for more information.

Release History

Release 27.0

- Command introduced

Overview

`POWER PEARSON ONESAMPLE` invokes the power analysis for the one-sample Pearson correlation test.

Options

Test Assumptions
Specifies the keywords that control the test assumptions. The keywords specify the sample size, power for sample size estimation, null hypothesis value of the correlation parameter, alternative hypothesis value of the correlation parameter, test direction, bias adjustment, and the significance level of the type I error rate. For more information, see "PARAMETERS Subcommand " on page 156.

Two-Dimensional Plot
Controls the two-dimensional power estimation by charts.

Three-Dimensional Plot
Controls the three-dimensional power estimation by sample size chart, and the three-dimensional power by null hypothesis value chart. Also controls the vertical rotation degrees (clockwise from the left) and the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart.

Basic specification

- `POWER PEARSON ONESAMPLE`: Invokes the power analysis for the one-sample Pearson correlation test.

Subcommand order

- The subcommands can be named in any order.

Example

```
POWER PEARSON ONESAMPLE
  /PARAMETERS TEST=NONDIRECTIONAL ADJUST_BIAS=TRUE SIGNIFICANCE=0.05
       N=4 NULL=1 ALTERNATIVE=.5
  /PLOT DIFF N_YAXIS DIFF_XAXIS VROTATE(10) HROTATE(325).
```

- **PARAMETERS** specifies that the test direction is non-directional, bias adjustment is enabled, the significance level uses the default 0.05 value, the sample size is set to 4, the null hypothesis value of the correlation parameter is set to 1, and the alternative hypothesis value of the correlation parameter is set to 0.5.
- **PLOT** specifies that the two-dimensional power by difference between hypothesized values chart displays in the output. The three-dimensional power by sample size (y-axis) and difference between hypothesized values (x-axis) chart also displays in the output. The vertical and horizontal rotation settings use their default values (10 degrees and 325 degrees respectively).

PARAMETERS Subcommand

PARAMETERS controls the test assumptions. The keywords specify the sample size, total number of trials for groups, proportion parameters for groups, significance level, statistical tests, estimation method, and test direction.

TEST = {NONDIRECTIONAL}{DIRECTIONAL}
An optional keyword that determines whether the test is one or two-sided. **NONDIRECTIONAL**, or a two-sided test, is the default setting. When **DIRECTIONAL** is specified, power is computed for a one-sided test.

ADJUST_BIAS = boolean
An optional keyword that determines whether the bias adjustment is involved or ignored. **TRUE** is the default setting, which includes the bias adjustment term in the power estimation. When **FALSE** is specified, the bias adjustment term is ignored.

SIGNIFICANCE = value
An optional keyword that specifies the significance level of the Type I error rate for the test. The value must be a single double value ∈`(0,1)`. The default level is `0.05`.

N = integer
A required keyword that specifies the sample size. The value must be a single integer > 3. **N** and **POWER** cannot be specified simultaneously.

POWER = value
A required keyword that specifies the power for sample size estimation. The value must be a single value ∈`(0,1)`. **POWER** and **N** cannot be specified simultaneously.

NULL = value
An optional keyword that specifies the null hypothesis value of the correlation parameter. Must be a single numeric value ∈`(-1,1)`. The default setting is 0. When **POWER** is specified, the **NULL** value cannot be -1 or 1.

ALTERNATIVE = value
A required keyword that specifies the alternative hypothesis value of the correlation parameter. Must be a single numeric value ∈`(-1,1)`. When **POWER** is specified, the **ALTERNATIVE** value cannot be -1 or 1, or equal to the **NULL** value.

PLOT Subcommand

The optional PLOT subcommand controls the output charts. By default, no charts are created. When the subcommand is specified, only the outputs for the following keywords are created.

N(integer1 integer2)
The optional keyword controls the two-dimensional power by sample size chart. The chart is suppressed by default. When specified, the chart displays and the plot boundaries are determined by **integer1** and **integer2**, respectively. When no integers are specified, the default plot range is used. Both **integer1** and **integer2** are integers, separated by a blank space (or blank spaces), and must satisfy: `4 ≤ integer1 < integer2 ≤ 5000`. You can either omit or specify both integer values.

NULL
The optional keyword controls the two-dimensional power by null value chart. The chart is suppressed by default. When specified, the chart displays.

ALTERNATIVE
The optional keyword controls the two-dimensional power by alternative value chart. The chart is suppressed by default. When specified, the chart displays.

DIFF
The optional keyword controls the two-dimensional power by difference between hypothesized values chart. The chart is suppressed by default. When specified, the chart displays.

N_XAXIS(integer1 integer2) DIFF_YAXIS

The optional keyword controls the three-dimensional power by sample size (x-axis) and difference between hypothesized values (y-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

Note: **`integer1`**and **`integer2`** follow the same rules as specified for **`N`**.

N_YAXIS(integer1 integer2) DIFF_XAXIS

The optional keyword controls the three-dimensional power by sample size (y-axis) and difference between hypothesized values (x-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

Note: **`integer1`**and **`integer2`** follow the same rules as specified for **`N`**.

NULL_XAXIS ALTERNATIVE_YAXIS

The optional keyword controls the three-dimensional power by null (x -axis) and alternative value (y-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

NULL_YAXIS ALTERNATIVE_XAXIS

The optional keyword controls the three-dimensional power by null (y-axis) and alternative value (x -axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

VROTATE(integer)

The optional keyword sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer ∈`[0,359]` with `10` as the default setting.

HROTATE(integer)

The optional keyword sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer ∈`[0,359]` with 325 as the default setting.

POWER PROPORTIONS INDEPENDENT

`POWER PROPORTIONS INDEPENDENT` is available in IBM SPSS Statistics Base Edition. The command invokes the power analysis for the independent-sample binomial test to compare two independent proportion parameters.

```
POWER PROPORTIONS INDEPENDENT
    /PARAMETERS
        [ TEST = {NONDIRECTIONAL**} {DIRECTIONAL} ]
        [ SIGNIFICANCE = {0.05**} {value} ]
        { N = {integer1 integer2} {POWER = {value}} }
        [ NRATIO = {1**} {value} ]
        PROPORTIONS = {value1 value2}
        [ METHOD = {CHISQ**} {T} {LRT} {FISHER} ]
        [ ESTIMATE = {NORMAL**} {BINOMIAL} ]
        [ CONTINUITY = {FALSE**} {TRUE} ]
        [ POOLED = {TRUE**} {FALSE} ]
    [ /PLOT
        [ N1_XAXIS([integer1 integer2]) N2_YAXIS([integer1 integer2]) ]
        [ N1_YAXIS([integer1 integer2]) N2_XAXIS([integer1 integer2]) ]
        [ P1_XAXIS P2_YAXIS ]
        [ P1_YAXIS P2_XAXIS ]
        [ NRATIO([value1 value2]) ]
        [ RISK_DIFF ]
        [ RISK_RATIO([value1 value2]) ]
        [ ODDS_RATIO([value1 value2]) ]
        [ VROTATE({10**} {integer}) ]
        [ HROTATE({325**} {integer}) ] ]
```

** Default if the subcommand or keyword is omitted.

This command reads the active dataset and causes execution of any pending commands. See the topic Command Order for more information.

Release History

Release 27.0

- Command introduced

Overview

`POWER PROPORTIONS INDEPENDENT` invokes power analysis for the independent-sample binomial test.

Options

Test Assumptions
Specifies the keywords that control the test assumptions. The keywords specify the sample size, total number of trials for groups, proportion parameters for groups, significance level, tests, estimation method, and test direction. For more information, see "PARAMETERS Subcommand " on page 158.

Two-Dimensional Plot
Controls the two-dimensional power estimation by charts.

Three-Dimensional Plot
Controls the three-dimensional power estimation by sample chart, and the three-dimensional power by null hypothesis value chart. Also controls the vertical rotation degrees (clockwise from the left) and the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart.

Basic specification

- `POWER PROPORTIONS INDEPENDENT`: Invokes power analysis for the independent-sample binomial test.

Subcommand order

- The subcommands can be named in any order.

Example

```
POWER PROPORTIONS INDEPENDENT
  /PARAMETERS TEST=NONDIRECTIONAL SIGNIFICANCE=0.05 N=1 2
      PROPORTIONS=.5 .75 METHOD=CHISQ ESTIMATE=BINOMIAL POOLED=TRUE
  /PLOT  N1_YAXIS  N2_XAXIS RISK_DIFF VROTATE(10) HROTATE(325).
```

- **PARAMETERS** specifies that the test direction is non-directional, the significance level uses the default 0.05 value, the total number of trials for the two groups is set to 1 and 2, the statistical test is chi-square, the estimation method is binomial approximation, and the estimation of the standard deviation is pooled.
- **PLOT** specifies that the three-dimensional power by number of trials in Group 2 (x -axis) and number of trials in Group 1 (y-axis) chart will display in the output. The two-dimensional power by risk ratio chart will also display in the output. The vertical and horizontal rotation settings use their default values (10 degrees and 325 degrees respectively).

PARAMETERS Subcommand

PARAMETERS controls the test assumptions. The keywords specify the sample size, total number of trials for groups, proportion parameters for groups, significance level, statistical tests, estimation method, and test direction.

TEST = {NONDIRECTIONAL}{DIRECTIONAL}
An optional keyword that determines whether the test is one or two-sided. **`NONDIRECTIONAL`**, or a two-sided test, is the default setting. When **`DIRECTIONAL`** is specified, power is computed for a one-sided test.

SIGNIFICANCE = value
An optional keyword that specifies the significance level of the Type I error rate for the test. The value must be a single double value ∈`(0,1)`. The default level is `0.05`.

N = (integer1 integer2)
A required keyword that specifies the total number of trials for two groups. Both integer values must be ≥ 1. When `METHOD = T`, `integer1 + integer2 > 2`.

POWER = value

A required keyword that specifies the power for sample size estimation. The value must be a single value ∈`(0,1)`. **POWER** and **N** cannot be specified simultaneously.

NRATIO = value

An optional keyword that specifies the sample size ratios. Must be a single value ∈`(0.01,100)` with 1 as the default setting. The keyword takes effect when **POWER** is specified and is ignored when **N** is specified.

PROPORTIONS = (value1 value2)

An optional keyword that specifies the proportion parameters for two groups. Both values must be ∈ `(0,1)`. **value1** and **value2** cannot be the same when **POWER** is specified.

METHOD = {CHISQ} {T} {LRT} {FISHER}

An optional keyword that specifies the statistical test. **CHISQ** estimates the power based on the Pearson's chi-squared test (the default setting). **T**, **LRT**, and **FISHER** estimate the power based on the Student's *t*-test, likelihood ratio test, and Fisher's exact test, respectively. Note that the combination of `METHOD = FISHER` and `ESTIMATE = BINOMIAL` takes effect when the sum of **integer1** and **integer2** specified for /**PARAMETERS N** does not exceed 500 (otherwise **CHISQ** is used).

Note: In some cases, `METHOD=FISHER` may take an extended amount of time to complete.

ESTIMATE = {NORMAL} {BINOMIAL}

An optional keyword that determines the power estimation method, if the total sample size specified for `/PARAMETERS N` does not exceed 500. **NORMAL** calls for the normal approximation (the default setting). **BINOMIAL** calls for the binomial enumeration method. The keyword has no effect when the specified `/PARAMETERS N` value exceeds 500 (normal approximation is used).

CONTINUITY = boolean

An optional keyword that controls whether or not continuity correction is used. **FALSE** is the default setting. Continuity correction is applied when **TRUE** is specified. The keyword takes effect when `METHOD = CHISQ`.

POOLED = boolean

An optional keyword that controls whether the estimation of the standard deviation is pooled or unpooled. **TRUE** is the default setting. When **FALSE** is specified, the unpooled standard deviation is calculated. The keyword takes effect when `METHOD = CHISQ` or `METHOD = T`.

PLOT Subcommand

The optional `PLOT` subcommand controls the output charts. By default, no charts are created. When the subcommand is specified, only the outputs for the following keywords are created.

N1_XAXIS(integer1 integer2) N2_YAXIS(integer1 integer2)

The optional keyword controls the three-dimensional power by number of trials in Group 1 (x -axis) and number of trials in Group 2 (y-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays. When no integers are specified, the default plot range is used. Both **integer1** and **integer2** are integers, separated by a blank space (or blank spaces). You can either omit or specify both integer values.

N1_YAXIS(integer1 integer2) N2_XAXIS(integer1 integer2)

The optional keyword controls the three-dimensional power by number of trials in Group 2 (x -axis) and number of trials in Group 1 (y-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays. When no integers are specified, the default plot range is used. Both **integer1** and **integer2** are integers, separated by a blank space (or blank spaces). You can either omit or specify both integer values.

P1_XAXIS P2_YAXIS

The optional keyword controls the three-dimensional power by proportion of Group 1 (x -axis) and proportion of Group 2 (y-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

P1_YAXIS P2_XAXIS

The optional keyword controls the three-dimensional power by proportion of Group 2 (x -axis) and proportion of Group 1 (y-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

NRATIO (value1 value2)

The optional keyword controls the two-dimensional power by group size ratio chart. The chart is suppressed by default. When the keyword is specified, the chart displays. When no values are specified, the default plot range is used. Both **`value1`** and **`value2`** are separated by a blank space (or blank spaces). You can either omit or specify both values.

RISK_DIFF

The optional keyword controls the two-dimensional power by risk difference chart. The chart is suppressed by default. When specified, the chart displays.

RISK_RATIO (value1 value2)

The optional keyword controls the two-dimensional power by risk ratio chart. The chart is suppressed by default. When specified, the chart displays. When no values are specified within the parentheses, the default plot range of the risk ratio is used. When specified, the plot boundaries are determined by **`value1`** and **`value2`**, respectively. Both **`value1`** and **`value2`** are separated by a blank space (or blank spaces). You can either omit or specify both values.

ODDS_RATIO (value1 value2)

The optional keyword controls the two-dimensional power by odds ratio chart. The chart is suppressed by default. When specified, the chart displays. When no values are specified within the parentheses, the default plot range of the odds ratio is used. When specified, the plot boundaries are determined by **`value1`** and **`value2`**, respectively. Both **`value1`** and **`value2`** are separated by a blank space (or blank spaces). You can either omit or specify both values.

VROTATE(integer)

The optional keyword sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer ∈`[0,359]` with `10` as the default setting.

HROTATE(integer)

The optional keyword sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer ∈`[0,359]` with 325 as the default setting.

POWER PROPORTIONS ONESAMPLE

`POWER PROPORTIONS ONESAMPLE` is available in IBM SPSS Statistics Base Edition. The command invokes power analysis for the one-sample binomial test to make statistical inference about the proportion parameter by comparing it with a hypothesized value.

```
POWER PROPORTIONS ONESAMPLE
   /PARAMETERS
      [ TEST = {NONDIRECTIONAL**} {DIRECTIONAL} ]
      [ ESTIMATE = {NORMAL**} {BINOMIAL} ]
      [ CONTINUITY = {FALSE**} {TRUE} ]
      [ SIGNIFICANCE = {0.05**} {value} ]
      { N = {integer} } { POWER = {value} }
      [ NULL = {0.50**} {value} ]
      ALTERNATIVE = {value}
   [ /PLOT
      [ N([integer1 integer2 ]) ] [ NULL ] [ ALTERNATIVE ] [ DIFF ]
      [ N_XAXIS([integer1 integer2]) DIFF_YAXIS ]
      [ N_YAXIS([integer1 integer2]) DIFF_XAXIS ]
      [ NULL_XAXIS ALTERNATIVE_YAXIS ]
      [ NULL_YAXIS ALTERNATIVE_XAXIS ]
      [ VROTATE({10**} {integer}) ]
      [ HROTATE({325**} {integer}) ] ]
```

** Default if the subcommand or keyword is omitted.

This command reads the active dataset and causes execution of any pending commands. See the topic Command Order for more information.

Release History

Release 27.0

- Command introduced

Overview

`POWER PROPORTIONS ONESAMPLE` invokes power analysis for the one-sample binomial test.

Options

Test Assumptions
Specifies the keywords that control the test assumptions. The keywords specify the sample size, total number of trials, sample size estimation, population proportion, null hypothesis value, estimation method, test direction, and significance level. For more information, see "PARAMETERS Subcommand " on page 161.

Two-Dimensional Plot
Controls the two-dimensional power by charts.

Three-Dimensional Plot
Controls the three-dimensional power estimation by total number of trials chart, and the three-dimensional power by null hypothesis value chart. Also controls the vertical rotation degrees (clockwise from the left) and the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart.

Basic specification

- `POWER PROPORTIONS ONESAMPLE`: Invokes power analysis for the one-sample binomial test.

Subcommand order

- The subcommands can be named in any order.

Example

```
POWER PROPORTIONS ONESAMPLE
  /PARAMETERS TEST=NONDIRECTIONAL ESTIMATE=NORMAL CONTINUITY=FALSE
       SIGNIFICANCE=0.05 N=2 NULL=.5 ALTERNATIVE=1
  /PLOT NULL ALTERNATIVE N_XAXIS DIFF_YAXIS VROTATE(10)
       HROTATE(325).
```

- **PARAMETERS** specifies that the test direction is non-directional, the estimation method is normal approximation, continuity correction is disabled, the significance level uses the default 0.05 value, the total number of trials is set to 2, the null value uses the default value of 0.5, and the population proportion is set to 1.
- **PLOT** specifies that the two dimensional Power estimation versus null hypothesis value and the Power estimation versus alternative hypothesis value charts will display in the output. The Power estimation versus total number of trials on x-axis and the difference between hypothesized values on y-axis three dimensional chart will also display in the output. The vertical and horizontal rotation settings use their default values (10 degrees and 325 degrees respectively).

PARAMETERS Subcommand

PARAMETERS controls the test assumptions. The keywords specify the sample size, total number of trials, sample size estimation, population proportion, null hypothesis value, estimation method, test direction, and significance level.

TEST = {NONDIRECTIONAL}{DIRECTIONAL}
An optional keyword that determines whether the test is one or two-sided. **NONDIRECTIONAL**, or a two-sided test, is the default setting. When **DIRECTIONAL** is specified, power is computed for a one-sided test.

ESTIMATE = {NORMAL} {BINOMIAL}
An optional keyword that determines the power estimation method, if the total sample size specified for `/PARAMETERS N` does not exceed `500`. **`NORMAL`** calls for the normal approximation (the default setting). **`BINOMIAL`** calls for the binomial enumeration method. The keyword has no effect when the specified `/PARAMETERS N` value exceeds `500` (normal approximation is used).

CONTINUITY = boolean
An optional keyword that controls whether or not the continuity correction is used for the normal approximation method. **`FALSE`** is the default setting. Continuity correction is applied when **`TRUE`** is specified. The keyword takes effect when `ESTIMATE = NORMAL`.

SIGNIFICANCE = value
An optional keyword that specifies the significance level of the Type I error rate for the test. The value must be a single double value ∈`(0,1)`. The default level is `0.05`.

N = integer
A required keyword that specifies the sample size. The value must be a single integer ≥ `1`.

POWER = value
A required keyword that specifies the power for sample size estimation. The value must be a single value ∈`(0,1)`. **`POWER`** and **`N`** cannot be specified simultaneously.

NULL = value
An optional keyword that specifies the null hypothesis value of the proportion parameter. Must be a single numeric value ∈`(0,1)`. The default setting is 0.50.

ALTERNATIVE = value
A required keyword that specifies the alternative hypothesis value of the proportion parameter. Must be a single numeric value ∈`(0,1)`. When **`POWER`** is specified, the **`ALTERNATIVE`** value cannot be equal to the **`NULL`** value, and must be a single numeric value ∈`(0,1)`.

PLOT Subcommand

The optional `PLOT` subcommand controls the output charts. By default, no charts are created. When the subcommand is specified, only the outputs for the following keywords are created.

N(integer1 integer2)
The optional keyword controls the two-dimensional power by total number of trials chart. The chart is suppressed by default. When specified, the chart displays and the plot boundaries are determined by **`integer1`** and **`integer2`**, respectively. When no integers are specified, the default plot range is used. Both **`integer1`** and **`integer2`** are integers, separated by a blank space (or blank spaces), and must satisfy: `0 < integer1 < integer2 ≤ 5000`. You can either omit or specify both integer values.

NULL
The optional keyword controls the two-dimensional power by null value chart. The chart is suppressed by default. When specified, the chart displays.

ALTERNATIVE
The optional keyword controls the two-dimensional power by alternative value chart. The chart is suppressed by default. When specified, the chart displays.

DIFF
The optional keyword controls the two-dimensional power by risk difference chart. The chart is suppressed by default. When specified, the chart displays.

N_XAXIS(integer1 integer2) DIFF_YAXIS
The optional keyword controls the three-dimensional power by total number of trials (x -axis) and risk difference (y-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

Note: `integer1`and **`integer2`** follow the same rules as specified for **`N`**.

N_YAXIS(integer1 integer2) DIFF_XAXIS

The optional keyword controls the three-dimensional power by total number of trials (y-axis) and risk difference (x -axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

Note: **`integer1`**and **`integer2`** follow the same rules as specified for **`N`**.

NULL_XAXIS ALTERNATIVE_YAXIS

The optional keyword controls the three-dimensional power by null (x -axis) and alternative value (y-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

NULL_YAXIS ALTERNATIVE_XAXIS

The optional keyword controls the three-dimensional power by null (y-axis) and alternative value (x -axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

VROTATE(integer)

The optional keyword sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer ∈`[0,359]` with `10` as the default setting.

HROTATE(integer)

The optional keyword sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer ∈`[0,359]` with 325 as the default setting.

POWER PROPORTIONS RELATED

`POWER PROPORTIONS RELATED` is available in IBM SPSS Statistics Base Edition. The command invokes power analysis for the related-sample binomial test (or McNemar's test) to compare two proportion parameters based on the matched pair subjects sampled from two related binomial populations.

```
POWER PROPORTIONS RELATED
    /PARAMETERS
        [ TEST = {NONDIRECTIONAL**} {DIRECTIONAL} ]
        [ ESTIMATE = {NORMAL**} {BINOMIAL} ]
        [ SIGNIFICANCE = {0.05**} {value} ]
        { NPAIRS = {integer} } { POWER = {value} }
        VALUES = {PROPORTIONS({value1 value2})} {COUNTS({integer1 integer2})}
        [ MARGINAL = {FALSE**} {TRUE} ]
        CORRELATION = {value}
    [ /PLOT
        [ NPAIRS([integer1 integer2]) ]
        [ RISK_DIFF ]
        [ RISK_RATIO([value1 value2]) ]
        [ ODDS_RATIO([value1 value2]) ]
        [ CORRELATION ]
        [ DISCORDANT ]
        [ MARGINAL ]
        [ VROTATE({10**} {integer}) ]
        [ HROTATE({325**} {integer}) ] ]
```

** Default if the subcommand or keyword is omitted.

This command reads the active dataset and causes execution of any pending commands. See the topic Command Order for more information.

Release History

Release 27.0

- Command introduced

Overview

`POWER PROPORTIONS RELATED` invokes power analysis for the related-sample binomial test (or McNemar's test).

Options

Test Assumptions

Specifies the keywords that control the test assumptions. The keywords specify the sample size,test values for proportions or counts, marginal test values, estimation method, test direction, and significance level. For more information, see "PARAMETERS Subcommand " on page 164.

Two-Dimensional Plot

Controls the two-dimensional power by charts.

Three-Dimensional Plot

Controls the three-dimensional power estimation by discordant proportions chart, and the three-dimensional power by marginal proportions chart. Also controls the vertical rotation degrees (clockwise from the left) and the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart.

Basic specification

- `POWER PROPORTIONS RELATED`: Invokes power analysis for the related-sample binomial test (or McNemar's test).

Subcommand order

- The subcommands can be named in any order.

Example

```
POWER PROPORTIONS RELATED
  /PARAMETERS TEST=NONDIRECTIONAL SIGNIFICANCE=0.05 NPAIRS=2
      VALUES=PROPORTIONS(.5 .75) MARGINAL=TRUE CORRELATION=0 ESTIMATE=NORMAL
  /PLOT RISK_DIFF DISCORDANT VROTATE(10) HROTATE(325).
```

- **PARAMETERS** specifies that the test direction is non-directional, the significance level uses the default 0.05 value, the total number of pairs is set to 2, the testing values for proportions are set the 0.5 and 0.75, marginal test values are enabled, the correlation between matched pairs is set to 0, and the estimation method is normal approximation.
- **PLOT** specifies that the two-dimensional Power estimation versus risk difference chart will display in the output. The three-dimensional Power estimation versus discordant proportions chart will also display in the output. The vertical and horizontal rotation settings use their default values (10 degrees and 325 degrees respectively).

PARAMETERS Subcommand

PARAMETERS controls the test assumptions. The keywords specify the sample size,test values for proportions or counts, marginal test values, estimation method, test direction, and significance level.

TEST = {NONDIRECTIONAL}{DIRECTIONAL}

An optional keyword that determines whether the test is one or two-sided. **NONDIRECTIONAL**, or a two-sided test, is the default setting. When **DIRECTIONAL** is specified, power is computed for a one-sided test.

ESTIMATE = {NORMAL} {BINOMIAL}

An optional keyword that determines the power estimation method, if the total sample size specified for `/PARAMETERS N` does not exceed `500`. **NORMAL** calls for the normal approximation (the default setting). **BINOMIAL** calls for the binomial enumeration method. The keyword has no effect when the specified `/PARAMETERS N` value exceeds `500` (normal approximation is used).

SIGNIFICANCE = value

An optional keyword that specifies the significance level of the Type I error rate for the test. The value must be a single double value `∈(0,1)`. The default level is `0.05`.

NPAIRS = integer

A required keyword that specifies the total number of the matched pairs. Must be a positive integer ≥ 2. **NPAIRS** and **POWER** cannot be specified simultaneously.

POWER = value
A required keyword that specifies the power for sample size estimation. The value must be a single value ∈`(0,1)`. **`POWER`** and **`NPAIRS`** cannot be specified simultaneously.

VALUES = {PROPORTIONS({value1 value2 })} {COUNTS({integer1 integer2 })}
A required keyword that specifies the proportions or counts. Only **`PROPORTIONS`** is supported when **`POWER`** is specified.

MARGINAL = boolean
An optional keyword that controls whether or not the proportions or the counts specified for **`VALUES`** are marginal. **`FALSE`** is the default setting, which denotes the discordant proportions or counts. When **`TRUE`** is specified, it denotes the marginal proportions or counts.

CORRELATION = value
A required keyword that specifies the correlation between matched pairs, which takes effect when `MARGINAL = TRUE`. The keyword is ignored when MARGINAL = FALSE. Must be a single value ∈ `(-1,1)`.

PLOT Subcommand

The optional PLOT subcommand controls the output charts. By default, no charts are created. When the subcommand is specified, only the outputs for the following keywords are created.

NPAIRS(integer1 integer2)
The optional keyword controls the two-dimensional power by total matched pairs chart. The chart is suppressed by default. When specified, the chart displays and the plot boundaries are determined by **`integer1`** and **`integer2`**, respectively. When no integers are specified, the default plot range is used. Both **`integer1`** and **`integer2`** are integers, separated by a blank space (or blank spaces), and must satisfy: `1 < integer1 < integer2 ≤ 2500`. You can either omit or specify both integer values.

RISK_DIFF
The optional keyword controls the two-dimensional power by risk difference chart. The chart is suppressed by default. When specified, the chart displays.

RISK_RATIO (value1 value2)
The optional keyword controls the two-dimensional power by risk ratio chart. The chart is suppressed by default. When specified, the chart displays. When no values are specified within the parentheses, the default plot range of the risk ratio is used. When specified, the plot boundaries are determined by **`value1`** and **`value2`**, respectively. Both **`value1`** and **`value2`** are separated by a blank space (or blank spaces). You can either omit or specify both values.

ODDS_RATIO (value1 value2)
The optional keyword controls the two-dimensional power by odds ratio chart. The chart is suppressed by default. When specified, the chart displays. When no values are specified within the parentheses, the default plot range of the odds ratio is used. When specified, the plot boundaries are determined by **`value1`** and **`value2`**, respectively. Both **`value1`** and **`value2`** are separated by a blank space (or blank spaces). You can either omit or specify both values.

CORRELATION
The optional keyword controls the two-dimensional power by correlation chart. The chart is suppressed by default. When specified, the chart displays. The keyword takes effect when `MARGINAL = TRUE`. The keyword is ignored when `MARGINAL = FALSE`.

DISCORDANT
The optional keyword controls the three-dimensional power by discordant proportions chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

MARGINAL
The optional keyword controls the three-dimensional power by marginal proportions chart. The chart is suppressed by default. When the keyword is specified, the chart displays. The keyword takes effect when `MARGINAL = TRUE`. The keyword is ignored when `MARGINAL = FALSE`.

VROTATE(integer)

The optional keyword sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer ∈`[0,359]` with `10` as the default setting.

HROTATE(integer)

The optional keyword sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer ∈`[0,359]` with `325` as the default setting.

POWER SPEARMAN ONESAMPLE

`POWER SPEARMAN ONESAMPLE` is available in IBM SPSS Statistics Base Edition. The command invokes the power analysis for the one-sample Spearman rank-order correlation test. Spearman rank-order correlation coefficient is a rank-based nonparametric statistic to measure the monotonic relationship between two variables that are usually censored and not normally distributed.

```
POWER SPEARMAN ONESAMPLE
    /PARAMETERS
        [ TEST = {NONDIRECTIONAL**} {DIRECTIONAL} ]
        [ VARIANCE = {BW**} {FHP} {CC} ]
        [ SIGNIFICANCE = {0.05**} {value} ]
        { N = {integer} } { POWER = {value} }
        [ NULL = {0**} {value} ]
        ALTERNATIVE = {value}
    [ /PLOT
        [ N([integer1 integer2 ]) ] [ NULL ] [ ALTERNATIVE ] [ DIFF ]
        [ N_XAXIS([integer1 integer2]) DIFF_YAXIS ]
        [ N_YAXIS([integer1 integer2]) DIFF_XAXIS ]
        [ NULL_XAXIS ALTERNATIVE_YAXIS ]
        [ NULL_YAXIS ALTERNATIVE_XAXIS ]
        [ VROTATE({10**} {integer}) ]
        [ HROTATE({325**} {integer}) ] ]
```

** Default if the subcommand or keyword is omitted.

This command reads the active dataset and causes execution of any pending commands. See the topic Command Order for more information.

Release History

Release 27.0

- Command introduced

Overview

`POWER SPEARMAN ONESAMPLE` invokes the power analysis for the one-sample Spearman rank-order correlation test.

Options

Test Assumptions

Specifies the keywords that control the test assumptions. The keywords specify the sample size, power for sample size estimation, asymptotic variance, null hypothesis value of the correlation parameter, alternative hypothesis value of the correlation parameter, test direction, and the significance level of the type I error rate. For more information, see “PARAMETERS Subcommand ” on page 167.

Two-Dimensional Plot

Controls the two-dimensional power estimation by charts.

Three-Dimensional Plot

Controls the three-dimensional power estimation by charts. Also controls the vertical rotation degrees (clockwise from the left) and the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart.

Basic specification

- `POWER SPEARMAN ONESAMPLE`: Invokes the power analysis for the one-sample Spearman rank-order correlation test.

Subcommand order

- The subcommands can be named in any order.

Example

```
POWER SPEARMAN ONESAMPLE
  /PARAMETERS TEST=NONDIRECTIONAL SIGNIFICANCE=0.05
      VARIANCE=FHP N=4 NULL=1 ALTERNATIVE=1
  /PLOT DIFF NULL_XAXIS ALTERNATIVE_YAXIS VROTATE(10)
      HROTATE(325).
```

- **PARAMETERS** specifies that the test direction is non-directional, the significance level uses the default 0.05 value, Fieller, Hartley and Pearson (FHP) is specified as the asymptotic variance, the sample size is set to 4, the null hypothesis value of the correlation parameter is set to 1, and the alternative hypothesis value of the correlation parameter is set to 1.
- **PLOT** specifies that the two-dimensional power by difference between hypothesized values chart displays in the output. The three-dimensional power by sample size (x-axis) and difference between hypothesized values (y-axis) chart also displays in the output. The vertical and horizontal rotation settings use their default values (10 degrees and 325 degrees respectively).

PARAMETERS Subcommand

PARAMETERS controls the test assumptions. The keywords specify the sample size, power for sample size estimation, asymptotic variance, null hypothesis value of the correlation parameter, alternative hypothesis value of the correlation parameter, test direction, and the significance level of the type I error rate.

TEST = {NONDIRECTIONAL}{DIRECTIONAL}

An optional keyword that determines whether the test is one or two-sided. **NONDIRECTIONAL**, or a two-sided test, is the default setting. When **DIRECTIONAL** is specified, power is computed for a one-sided test.

Variance = {BW} {FHP} {CC}

An optional keyword that determines how the asymptotic variance is estimated for the power analysis. **BW** is the setting by default, which estimates the variance suggested

by Bonett and Wright. **FHP** and **CC** estimate the variance suggested by Fieller et al.,and Caruso and Cliff, respectively.

SIGNIFICANCE = value

An optional keyword that specifies the significance level of the Type I error rate for the test. The value must be a single double value ∈`(0,1)`. The default level is `0.05`.

N = integer

A required keyword that specifies the sample size. The value must be a single integer > 3 when `VARIANCE = BW` or FHP, and > 2 when `VARIANCE = CC`. **N** and **POWER** cannot be specified simultaneously.

POWER = value

A required keyword that specifies the power for sample size estimation. The value must be a single value ∈`(0,1)`. **POWER** and **N** cannot be specified simultaneously.

NULL = value

An optional keyword that specifies the null hypothesis value of the correlation parameter. Must be a single numeric value ∈`(-1,1)`. The default setting is 0.

ALTERNATIVE = value

A required keyword that specifies the alternative hypothesis value of the correlation parameter. Must be a single numeric value ∈`(-1,1)`. When **`POWER`** is specified, the **`ALTERNATIVE`** value cannot be equal to the **`NULL`** value.

PLOT Subcommand

The optional PLOT subcommand controls the output charts. By default, no charts are created. When the subcommand is specified, only the outputs for the following keywords are created.

N(integer1 integer2)

The optional keyword controls the two-dimensional power by total number of trials chart. The chart is suppressed by default. When specified, the chart displays and the plot boundaries are determined by **`integer1`** and **`integer2`**, respectively. When no integers are specified, the default plot range is used. Both **`integer1`** and **`integer2`** are integers, separated by a blank space (or blank spaces), and must satisfy: `4 ≤ integer1 < integer2 ≤ 5000`. You can either omit or specify both integer values.

NULL

The optional keyword controls the two-dimensional power by null value chart. The chart is suppressed by default. When specified, the chart displays.

ALTERNATIVE

The optional keyword controls the two-dimensional power by alternative value chart. The chart is suppressed by default. When specified, the chart displays.

DIFF

The optional keyword controls the two-dimensional power by risk difference chart. The chart is suppressed by default. When specified, the chart displays.

N_XAXIS(integer1 integer2) DIFF_YAXIS

The optional keyword controls the three-dimensional power by total number of trials (x -axis) and risk difference (y-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

Note: `integer1`and **`integer2`** follow the same rules as specified for **`N`**.

N_YAXIS(integer1 integer2) DIFF_XAXIS

The optional keyword controls the three-dimensional power by total number of trials (y-axis) and risk difference (x -axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

Note: `integer1`and **`integer2`** follow the same rules as specified for **`N`**.

NULL_XAXIS ALTERNATIVE_YAXIS

The optional keyword controls the three-dimensional power by null (x -axis) and alternative value (y-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

NULL_YAXIS ALTERNATIVE_XAXIS

The optional keyword controls the three-dimensional power by null (y-axis) and alternative value (x -axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

VROTATE(integer)

The optional keyword sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer ∈`[0,359]` with `10` as the default setting.

HROTATE(integer)

The optional keyword sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer ∈`[0,359]` with 325 as the default setting.

POWER UNIVARIATE LINEAR

POWER UNIVARIATE LINEAR is available in IBM SPSS Statistics Base Edition. The command invokes power analysis for the type III *F*-test in univariate linear regression. Univariate linear regression is a basic and standard statistical approach in which researchers use the values of several variables to explain or predict values of a scale outcome.

```
POWER UNIVARIATE LINEAR
    /PARAMETERS
        [ MODEL ={FIXED**} {RANDOM}]
        [ SIGNIFICANCE = {0.05**} {value}]
        {N ={integer} } {POWER ={value} }
        TOTAL_PREDICTORS = {integer}
        TEST_PREDICTORS = {integer}
        {PARTIAL_CORR = {value} } {FULL_MODEL = {value} NESTED_MODEL ={value} }
        [ INTERCEPT = {TRUE**} {FALSE}]
    [ /PLOT
        [ N([integer1 integer2]) ]
        [ PARTIAL_CORR ]
        [ N_XAXIS([integer1 integer2]) PARTIAL_CORR_YAXIS ]
        [ N_YAXIS([integer1 integer2]) PARTIAL_CORR_XAXIS ]
        [ VROTATE({10**} {integer}) ]
        [ HROTATE({325**} {integer}) ] ]
```

** Default if the subcommand or keyword is omitted.

This command reads the active dataset and causes execution of any pending commands. See the topic Command Order for more information.

Release History

Release 27.0

- Command introduced

Overview

`POWER UNIVARIATE LINEAR` invokes power analysis for the type III *F*-test in univariate linear regression. Univariate linear regression is a basic and standard statistical approach in which researchers use the values of several variables to explain or predict values of a scale outcome.

Options

Test Assumptions

Specifies the keywords that control the test assumptions. The keywords specify the sample size, the power for sample size estimation, the population multiple partial correlation, number of either the total predictors or the predictors in the full model, number of either the test predictors or the predictors in the nested mode, the multiple partial correlation coefficient, the squared multiple correlation coefficients for the full and nested model, the intercept term, model predictors, and the significance level of the type I error rate for the test. For more information, see "PARAMETERS Subcommand " on page 170.

Two-Dimensional Plot

Controls the two-dimensional power by sample size chart, and the two-dimensional power by effect size chart.

Three-Dimensional Plot

Controls the three-dimensional power by sample size (x-axis) and effect size (y-axis) chart, and the three-dimensional power by sample size (y-axis) and effect size (x-axis) chart. Also controls the vertical rotation degrees (clockwise from the left) and the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart.

Basic specification

- `POWER UNIVARIATE LINEAR`: Invokes power analysis for the type III *F*-test in univariate linear regression.

Subcommand order

- The subcommands can be named in any order.

Example

```
POWER UNIVARIATE LINEAR
  /PARAMETERS MODEL=FIXED SIGNIFICANCE=0.05 N=5 TOTAL_PREDICTORS=2 TEST_PREDICTORS=2 PARTIAL_CORR
    =0.5 INTERCEPT =TRUE
  /PLOT PARTIAL_CORR N_XAXIS  PARTIAL_CORR_YAXIS VROTATE(10) HROTATE(325).
```

- **PARAMETERS** specifies that the model predictors are fixed, the significance level uses the default 0.05 value, the total number of predictors is 2, the total number of test predictors is 2, the multiple partial correlation coefficient is 0.5, and the intercept term is included.
- **PLOT** specifies that the two-dimensional power by multiple partial correlation coefficient chart displays in the output. The three-dimensional power by sample size (x-axis) and multiple partial correlation coefficient (y-axis) chart also displays in the output. The vertical and horizontal rotation settings use their default values (10 degrees and 325 degrees respectively).

PARAMETERS Subcommand

PARAMETERS controls the test assumptions. The keywords specify the sample size, the power for sample size estimation, the population multiple partial correlation, number of either the total predictors or the predictors in the full model, number of either the test predictors or the predictors in the nested mode, the multiple partial correlation coefficient, the squared multiple cor-relation coefficients for the full and nested model, the intercept term, model predictors, and the significance level of the type I error rate for the test.

MODEL = {FIXED}{RANDOM}

An optional keyword that determines whether the predictors are assumed to be fixed or random. **FIXED** is the default setting, which assumes that the predictors have fixed values. When **RANDOM** is specified, the predictors are assumed to be random.

SIGNIFICANCE = value

An optional keyword that specifies the significance level of the Type I error rate for the test. The value must be a single double value ∈`(0,1)`. The default level is `0.05`.

N = integer

A required keyword that specifies the sample size. Must be a single integer ≥ `(p + 2)` when `INTERCEPT = TRUE`, and ≥ `(p + 1)` in all other conditions (*p* is the value specified for **TOTAL_PREDICTORS**. **N** and **POWER** cannot be specified simultaneously.

POWER = value

A required keyword that specifies the power for sample size estimation. The value must be a single value ∈`(0,1)`. POWER and N cannot be specified simultaneously.

TOTAL_PREDICTORS = integer

A required keyword that specifies the number of either the total predictors or the predictors in the full model (not including the intercept if applicable). Must be a single integer ≥ 1.

TEST_PREDICTORS = integer

A required keyword that specifies the number of either the test predictors or the predictors in the nested model (not including the intercept if applicable). When **PARTIAL_CORR** is specified, the value must be a single integer ∈`[1,p]`, where p is the value specified for **TOTAL_PREDICTORS**. When **FULL_MODEL** and **NESTED_MODEL** are specified, the value must be a single integer ∈`[1,p)`, where *p* is the value specified for **TOTAL_PREDICTORS**.

PARTIAL_CORR = value

A required keyword that specifies the value of the multiple partial correlation coefficient. Must be a single value ∈`(-1,1)`. When **POWER** is specified, the value cannot be 0. **PARTIAL_CORR** and **FULL_MODEL**/**NESTED_MODEL** are mutually exclusive keywords, and only one is allowed.

FULL_MODEL = value NESTED_MODEL = value

A required keyword that specifies the squared multiple correlation coefficients for the full and nested models, respectively. **FULL_MODEL**/**NESTED_MODEL** and **PARTIAL_CORR** are mutually exclusive keywords, and only one is allowed. When specified, both **`FULL_MODEL`**/**`NESTED_MODEL`** are required, and a single value ∈`(0,1)` for each keyword is required. In addition, the values specified for **`FULL_MODEL`** must be greater than or equal to the value specified for **`NESTED_MODEL`**. When **`POWER`** is specified, the value specified for **`FULL_MODEL`** must be greater than the value specified for **`NESTED_MODEL`**.

INTERCEPT = boolean
An optional keyword that determines whether or not the intercept term is included. **`TRUE`** is the default setting, which includes the intercept term. When **`FALSE`** is specified, the intercept term is excluded from the power analysis.

PLOT Subcommand

The optional PLOT subcommand controls the output charts. By default, no charts are created. When the subcommand is specified, only the outputs for the following keywords are created.

N(integer1 integer2)
The optional keyword controls the two-dimensional power by sample size chart. By default, the chart is suppressed. When specified, the chart is displayed. When no integers are specified within the parentheses, the default plot range of the sample size is used. When the keyword is specified, the plot boundaries are determined by **`integer1`** and **`integer2`**, respectively. It is required that `p + 2 ≤ integer1 < integer2 ≤ p + 5000` (where *p* is the value specified for TOTAL_PREDICTORS) when `/PARAMETERS INTERCEPT = TRUE`. In all other scenarios, `p + 1 ≤ integer1 < integer2 ≤ p + 5000`. Note that both **`integer1`** and **`integer2`** are integers, separated by a blank space (or blank spaces). You can either omit or specify both values.

PARTIAL_CORR
The optional keyword controls the two-dimensional power by multiple partial correlation coefficient chart. The chart is suppressed by default. When the keyword is specified, the chart displays.

N_XAXIS(integer1 integer2) PARTIAL_CORR_YAXIS
The optional keyword controls the three-dimensional power by sample size (x-axis) and multiple partial correlation coefficient (y-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays. Note that **`integer1`** and **`integer2`** follow the same rules as specified for **`N`**.

N_YAXIS(integer1 integer2) PARTIAL_CORR_XAXIS
The optional keyword controls the three-dimensional power by sample size (y-axis) and multiple partial correlation coefficient (x-axis) chart. The chart is suppressed by default. When the keyword is specified, the chart displays. Note that **`integer1`** and **`integer2`** follow the same rules as specified for **`N`**.

VROTATE(integer)
The optional keyword sets the vertical rotation degrees (clockwise from the left) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer ∈`[0,359]` with `10` as the default setting.

HROTATE(integer)
The optional keyword sets the horizontal rotation degrees (clockwise from the front) for the three-dimensional chart. The keyword takes effect when the three-dimensional plot is requested. The value must be a single integer ∈`[0,359]` with `325` as the default setting.

QUANTILE REGRESSION

QUANTILE REGRESSION is available in SPSS Statistics Standard Edition or the Advanced Statistics Option.

```
QUANTILE REGRESSION dependent_varname [BY factor_list] [WITH covariate_list]
[ /MISSING
     [ CLASSMISSING = {EXCLUDE**} {INCLUDE}]]
[ /CRITERIA
     [ QUANTILE = {0.5**} {[values] [value1 TO value2 BY value3]} ]   @@
     [ IID = {TRUE**} {FALSE} ]
     [ BANDWIDTH = {BOFINGER**} {HALL_SHEATHER} ]
     [ CILEVEL = {95**} {value} ]
     [ METHOD = {AUTO**} {SIMPLEX} {INTERIOR_POINT} ]
```

```
      [ TOL = {1E-12**} {value} ]   ##
      [ CONV = {1E-6**} {value}]   $$
      [ MAXITER = {2000**} {integer} ] ]
 [ /MODEL
      [ effect_1 effect_2 ... ]
      [ REGWGT = weight_varname ]   %%
      [ INTERCEPT = {TRUE**} {FALSE} ] ]  ^^
 [ /SAVE
      [ PRED([ {varname} {rootname}) ] ]
      [ RESID( {varname} {rootname}) ]
      [ CIPREDL({varname} {rootname}) ]
      [ CIPREDU({varname} {rootname}) ] ]
 [ /PREDICT_EFFECTS
      [ NUM_TOP_EFFECTS = {3**} {integer}] ]
 [ /PREDICTION_TABLES ]
      [ effect_1 effect_2 ... ]   &&
 [ /PREDICTION_LINES ]
      [ effect_1 effect_2 ... ]   ⊕
 [ /PRINT
      [ PARAMETER** ]
      [ COVB ]
      [ CORB ] ]
 [ /PLOT
      [ MAX_CATEGORIES = {10**} {integer} ]   ⊖
      [ PREDICTED_BY_OBSERVED = {FALSE**} {TRUE} ]
      [ PARAM_EST = {50**} {integer} {ALL} ] ]
 [ /OUTFILE
      [ COVB = 'savfile' | 'dataset' ]
      [ CORB = 'savfile' | 'dataset' ]
      [ FILE_SEPARATE = {FALSE**} {TRUE} ]
      [ {MODEL = 'filename'} {PARAMETER = 'filename'} ] ]
 [ /EXTERNAL ]
```

** Default if the subcommand or keyword is omitted.

@@ Use unique values ∈ [0,1]

For the matrix manipulations

$$ For the numerical method

%% Must be a numeric variable

^^ Either intercept or at least one predictor

&& A subset of the effects contained in the model

⊕ A subset of the effects contained in the model

⊖ Maximum categories plotted for a mixed effect

Release History

Release 26.0

- Command introduced

Release 26.0 Fix Pack 1

- QUANTILE keyword support for grid of quantiles (TO and BY)

Overview

`QUANTILE REGRESSION dependent_varname [BY factor_list] [WITH covariate_list]` is a required command to invoke the quantile regression procedure. A single numeric dependent variable is required. BY and `WITH` are followed by an optional list of factors and covariates, respectively. Note that either the intercept term or at least one predictor is required to run the analysis.

Options

Analysis options

You can specify the regression percentile, the estimation method, numerical method settings, post-estimation attributes, and the significance level for computing credible intervals.

Model design

You can specify the model design for the analysis, which includes target variables, factor variables, covariates, regression weights, and model effects.

Display settings

You can specify which estimates to output, as well as plot and tabulation output options for the effects, prediction lines and tables.

Save settings
You can specify to save the predicted value of responses, residuals, and the lower/upper bound of prediction intervals.

Export options
You can specify to export the covariance and correlation matrices for parameter estimates, and specify a target file to generate the PMML file.

Basic specification

- `QUANTILE REGRESSION dependent_varname [BY factor_list] [WITH covariate_list]` is a required command to invoke the quantile regression procedure.

Subcommand order

- The subcommands can be named in any order.

Syntax rules

- In cases where duplicated values are specified, only unique values are used, and a warning message is generated to indicate that all duplicated values are excluded from the analysis.
- Only one regression weight variable is allowed.
- The specified **FACTORS** must be a subset of **`factor_list`**.
- The specified **COVARIATES** must be a subset of **`covariate_list`**.
- **BAYESPRED** can be **TRUE** if values are specified for **REGRESSORS**.
- **VARIABLES** must be a subset of **COVARIATES** and **FACTORS**.
- **INTERCEPT** must be **TRUE** when **VARIABLES** is empty.

CRITERIA Subcommand

`CRITERIA` controls the statistical criteria used in building the Bayesian Linear Regression models. The default criteria are noted in the description of each keyword below.

QUANTILE = value(s). *Optional keyword that specifies the quantile(s)* The single value `0.5` is the default setting. Users are allowed to specify multiple values with each value separated by a blank space (or blank spaces). Users are also allowed to specify a grid of quantiles from `value1` to `value2` with the increment of `value3`, connected by the keywords **TO** and **BY**. When a grid of quantiles is specified, only one valid set of `[value1 TO value2 BY value3]` is allowed. It must satisfy that `0 ≤ value1 ≤ value2 ≤ 1`. In cases where `value1 = value2`, it is equivalent to specifying a single `value1`, regardless of `value3`. In order to restrict the number of quantiles, `0.01 ≤ value3 ≤ 1`. The grid can be mixed with other quantiles, and can be placed anywhere.

Notes:

- All specified values must be in `[0,1]`.
- In cases where duplicated values are specified, only unique values are used, and a warning message is generated to indicate that all of the duplicated values are excluded from the analysis.

IID = boolean. *Determines whether or not the error terms are independently and identically distributed.* This is an option for the post-estimation of the variance-covariance of the parameter estimates and the confidence intervals for the predicted target values. The default setting is **TRUE**. When **FALSE** is specified, the computation time may be significantly increased for large models.

BANDWIDTH = value. *Determines which bandwidth method is used to estimate the variance-covariance matrix of the parameter estimates.* **BOFINGER** and **HALL_SHEATHER** are possible values (**BOFINGER** is the default value).

CILEVEL = value. *Specify the significance level for computing credible intervals.* Must be a single double value `∈(0,100)`. The default level is 95%

METHOD = value. *Specifies the model estimation method.* The **`SIMPLEX`** option calls for the simplex algorithm developed by Barrodale and Roberts. The **`INTERIOR-POINT`** option calls for the Frisch-Newton interior-point non-linear optimization algorithm. The **`AUTO`** option is the default setting, which allows the procedure to automatically choose the appropriate estimation method.

TOL = value. *Specifies the tolerance value for the matrix manipulations in the interior-point method.* `0.000001` is the default setting. Only in effect when `METHOD = AGL`. Must be one value $\in$ `(0, 10`$^{-3}$`]`.

CONV = value. *Specifies the convergence criterion for the numerical method.* Must be a single double value in `(0, 10`$^{-3}$`]` with `10`$^{-6}$ as the default value.

MAXITER = integer. *Specify the maximum number of iterations the AGL method can go through in its computations.* The integer `2000` is the default setting. You must specify a positive integer. Valid when `METHOD = AGL`.

MODEL Subcommand

`DESIGN` specifies the model design for the analysis.

REGWGT = variable. *Specify the regression weights.* The parameter is optional. Only one variable is allowed.

FACTORS = factors. *Specify categorical variables for the model.* Must be a subset of the factors in **`BAYES REGRESSION`**.

COVARIATES = covariates. *Specify scale variables for the model.* Must be a subset of the factors in **`BAYES REGRESSION`**.

SAVE Subcommand

SAVE specifies the statistics to be scored for the Bayesian prediction distribution. Score the statistics derived from the Bayesian predictions, which include to following:

- **`{MEANPRED(varname)}`**: Mean of the posterior predictive distribution.
- **`{VARIANCE(varname)}`**: Variance of the posterior predictive distribution.
- **`{MODE(varname)}`**: Mode of the posterior predictive distribution.
- **`{CIMEANPREDL(varname)}`**: Lower bound of the credible interval of the posterior predictive distribution.
- **`{CIMEANPREDU(varname)}`**: Upper bound of the credible interval of the posterior predictive distribution.

Corresponding variable names can also be assigned.

PREDICT_EFFECTS Subcommand

`PREDICT` specifies the regressors to generate predictive distributions.

REGRESSORS = values. *Specify an observed vector with the values for the regressors.* Each regressor is assigned one value or string (allow one case can be predicted). For factors, both values and strings are allowed.

PREDICTION_TABLES Subcommand

`PREDICT` specifies the regressors to generate predictive distributions.

REGRESSORS = values. *Specify an observed vector with the values for the regressors.* Each regressor is assigned one value or string (allow one case can be predicted). For factors, both values and strings are allowed.

PREDICTION_LINES Subcommand

`PREDICT` specifies the regressors to generate predictive distributions.

REGRESSORS = values. *Specify an observed vector with the values for the regressors.* Each regressor is assigned one value or string (allow one case can be predicted). For factors, both values and strings are allowed.

PRINT Subcommand

PLOT Subcommand

PLOT controls the plots to be output.

COVARIATES = {covariate_sublist}. *Specify the covariate(s) to be plotted.* Must be a sublist of the covariates specified in **/DESIGN**.

FACTORS = {factor_sublist}. *Specify the factor(s) to be plotted.* Must be a sublist of the factors specified in **/DESIGN**.

MAX_CATEGORY = integer. *Specify the maximum factor levels to be plotted.* Applies to all factors. By default, the first 2 levels are plotted for each factor. Only a single, positive integer is valid.

INTERCEPT. *Specify whether the intercept term is plotted.* **FALSE** is the default setting.

ERRORVAR. *Specify whether the variance of errors is plotted.* **FALSE** is the default setting.

BAYESPRED. *Specify whether the predictive distribution is plotted.* **FALSE** is the default setting. The value can be **TRUE** only when valid values are specified for **REGRESSORS**.

OUTFILE Subcommand

OUTFILE specifies the target file to generate the PMML file.

PARAMETER = 'filename'. *Specify whether or not the variance-covariance matrix of the parameters will be scored in the output PMML file.*

EXTERNAL Subcommand

```
BAYES REGRESSION salary WITH educ prevexp
     /CRITERIA CILEVEL = 95
     /DESIGN COVARIATES = educ prevexp
     /INFERENCE ANALYSIS = BAYESFACTOR
     /PRIOR TYPE = REFERENCE .
```

- **CRITERIA** specifies that the significance level for computing credible intervals (**CILEVEL**) is set to the default value of 95.
- **DESIGN** specifies the model design for the analysis. The example uses the **educ** and **prevexp** as the scale variables.
- **INFERENCE** specifies the distribution for the analysis. The example uses Estimate Bayes Factor (**BAYESFACTOR**) as the **ANALYSIS** method.
- **PRIOR** specifies the prior distribution.

T-TEST

T-TEST is available in the Statistics Base Edition.

One-sample tests

```
T-TEST TESTVAL n /VARIABLE=varlist
```

Independent-samples tests

```
T-TEST GROUPS=varname ({1,2**      }) /VARIABLES=varlist
                       {value      }
                       {value,value}
```

**Default if the subcommand is omitted.

Paired-samples tests

```
T-TEST PAIRS=varlist [WITH varlist [(PAIRED)]] [/varlist ...]
[/ES
    [ DISPLAY({TRUE**} {FALSE}) ]
    [ STANDARDIZER({SD**} {CORRECTED_SD} {AVERAGE}) ] ]
```

All test types

```
[/MISSING={ANALYSIS**}  [INCLUDE]]
          {LISTWISE  }

[/CRITERIA=CI({0.95**})
              {value }
```

**Default if the subcommand is omitted.

Examples

```
T-TEST GROUPS=WORLD(1,3) /VARIABLES=NTCPRI NTCSAL NTCPUR.

T-TEST PAIRS=TEACHER CONSTRUC MANAGER.
```

Release History

Release 27.0

- **ES** subcommand introduced

Overview

`T-TEST` compares sample means by calculating Student’s *t* and displays the two-tailed probability of the difference between the means. Statistics are available for one-sample (tested against a specified value), independent samples (different groups of cases), or paired samples (different variables). Other procedures that compare group means are `ANOVA`, `ONEWAY`, `UNIANOVA`, `GLM`, and `MANOVA` (`GLM` and `MANOVA` are available in SPSS Statistics Standard Edition or the Advanced Statistics Option).

Options

Statistics. There are no optional statistics. All statistics available are displayed by default.

Basic Specification

The basic specification depends on whether you want a one-sample test, an independent-samples test or a paired-samples test. For all types of tests, `T-TEST` displays Student’s *t,* degrees of freedom, and two-tailed probabilities, as well as the mean, standard deviation, standard error, and count for each group or variable.

- To request a one-sample test, use the `TESTVAL` and `VARIABLES` subcommands. The output includes a One-Sample Statistics table showing univariate statistics and a One-Sample Test table showing the test value, the difference between the sample mean and the test value, and the two-tailed probability level.
- To request an independent-samples test, use the `GROUPS` and `VARIABLES` subcommands. The output includes a Group Statistics table showing summary statistics by group for each dependent variable and an Independent-Samples Test table showing both pooled- and separate-variance estimates, along with the *F* value used to test homogeneity of variance and its probability. The two-tailed probability is displayed for the *t* value.
- To request a paired-samples test, use the `PAIRS` subcommand. The output includes a Paired Statistics table showing univariate statistics by pairs, a Paired Samples Correlations table showing correlation coefficients and two-tailed probability level for a test of the coefficient for each pair, and a Paired Samples Test table showing the paired differences between the means and two-tailed probability levels for a test of the differences.

Subcommand Order

Subcommands can be named in any order.

Operations

- If a variable specified on `GROUPS` is a long string, only the short-string portion is used to identify groups in the analysis.
- Probability levels are two-tailed. To obtain the one-tailed probability, divide the two-tailed probability by 2.

Limitations

- Maximum of one `TESTVAL` and one `VARIABLES` subcommand per one-sample *t* test.
- Maximum of one `GROUPS` and one `VARIABLES` subcommand per independent-samples *t* test.

Examples

One-Sample Test

```
T-TEST
  /TESTVAL=322
  /VARIABLES=brake
  /CRITERIA=CIN (.90).
```

- The procedure compares the mean of *brake* with the test value 322.
- `CRITERIA` requests a 90% confidence interval for the mean difference.

Independent-Samples Test, using specific values

```
T-TEST GROUPS=insert(0 1)
  /MISSING=ANALYSIS
  /VARIABLES=dollars
  /CRITERIA=CIN(.95).
```

- The procedure compares the means of the two groups defined by values 0 and 1 of *insert* for variable *dollars*.
- All other options are set to their default values.

Paired-Samples Test

```
T-TEST PAIRS=TEACHER CONSTRUC MANAGER.
```

- This paired-samples *t* test compares the means of *TEACHER* with *CONSTRUC*, *TEACHER* with *MANAGER*, and *CONSTRUC* with *MANAGER*.

VARIABLES Subcommand

`VARIABLES` specifies the dependent variables to be tested in a one-sample or an independent-samples *t* test.

- `VARIABLES` can specify multiple variables, all of which must be numeric.
- When specified along with `TESTVAL`, the mean of all cases for each variable is compared with the specified value.
- When specified along with `GROUPS`, the means of two groups of cases defined by the `GROUPS` subcommand are compared.
- If both `TESTVAL` and `GROUPS` are specified, a one-sample test and an independent-samples test are performed on each variable.

TESTVAL Subcommand

`TESTVAL` specifies the value with which a sample mean is compared.

- Only one `TESTVAL` subcommand is allowed.
- Only one value can be specified on the `TESTVAL` subcommand.

GROUPS Subcommand

`GROUPS` specifies a variable used to group cases for independent-samples *t* tests.

- `GROUPS` can specify only one variable, which can be numeric or string.

Any one of three methods can be used to define the two groups for the variable specified on `GROUPS`:

- Specify a single value in parentheses to group all cases with a value equal to or greater than the specified value into one group and the remaining cases into the other group.
- Specify two values in parentheses to include cases with the first value in one group and cases with the second value in the other group. Cases with other values are excluded.
- If no values are specified on `GROUP`, `T-TEST` uses 1 and 2 as default values for numeric variables. There is no default for string variables.

PAIRS Subcommand

`PAIRS` requests paired-samples *t* tests.

- The minimum specification for a paired-samples test is `PAIRS` with an analysis list. Only numeric variables can be specified on the analysis list. The minimum analysis list is two variables.
- If keyword `WITH` is not specified, each variable in the list is compared with every other variable on the list.
- If keyword `WITH` is specified, every variable to the left of `WITH` is compared with every variable to the right of `WITH`. `WITH` can be used with `PAIRED` to obtain special pairing.
- To specify multiple analysis lists, use multiple `PAIRS` subcommands, each separated by a slash. Keyword `PAIRS` is required only for the first analysis list; a slash can be used to separate each additional analysis list.

(PAIRED). *Special pairing for paired-samples test.* `PAIRED` must be enclosed in parentheses and must be used with keyword `WITH`. When `PAIRED` is specified, the first variable before `WITH` is compared with the first variable after `WITH`, the second variable before `WITH` is compared with the second variable after `WITH`, and so forth. The same number of variables should be specified before and after `WITH`; unmatched variables are ignored and a warning message is issued. `PAIRED` generates an error message if keyword `WITH` is not specified on `PAIRS`.

Example

```
T-TEST  PAIRS=TEACHER CONSTRUC MANAGER.
T-TEST  PAIRS=TEACHER MANAGER WITH CONSTRUC ENGINEER.
T-TEST  PAIRS=TEACHER MANAGER WITH CONSTRUC ENGINEER (PAIRED).
```

- The first `T-TEST` compares *TEACHER* with *CONSTRUC, TEACHER* with *MANAGER*, and *CONSTRUC* with *MANAGER*.
- The second `T-TEST` compares *TEACHER* with *CONSTRUC*, *TEACHER* with *ENGINEER*, *MANAGER* with *CONSTRUC*, and *MANAGER* with *ENGINEER*. *TEACHER* is not compared with *MANAGER*, and *CONSTRUC* is not compared with *ENGINEER*.
- The third `T-TEST` compares *TEACHER* with *CONSTRUC* and *MANAGER* with *ENGINEER*.

CRITERIA Subcommand

`CRITERIA` resets the value of the confidence interval. Keyword `CI` is required. You can specify a value between 0 and 1 in the parentheses. The default is 0.95.

ES Subcommand

ES controls the estimation of the *t*-test effect size.

DISPLAY = boolean

An optional keyword that determines whether the effect size table is printed. **TRUE** is the default setting, which invokes the estimation of effect size. When **FALSE** is specified, effect size is not estimated, and the corresponding output table is suppressed.

STANDARDIZER = {SD} {CORRECTED_SD} {AVERAGE}

An optional keyword that controls how the standardizer is computed in estimating the Cohen's *d* and Hedges' correction for each variable pair. **SD** is the default setting (Standard deviation of the difference). The other options are **CORRECTED_SD** (corrected standard deviation of the difference) and **AVERAGE** (average of variances). The keyword takes effect only in the related-sample *t*-test, and it is ignored by the one-sample and independent-sample *t*-tests, or when `DISPLAY = FALSE`.

MISSING Subcommand

`MISSING` controls the treatment of missing values. The default is `ANALYSIS`.

- `ANALYSIS` and `LISTWISE` are alternatives; however, each can be specified with `INCLUDE`.

ANALYSIS. *Delete cases with missing values on an analysis-by-analysis or pair-by-pair basis.* For independent-samples tests, cases with missing values for either the grouping variable or the dependent variable are excluded from the analysis of that dependent variable. For paired-samples tests, a case with a missing value for either of the variables in a given pair is excluded from the analysis of that pair. This is the default.

LISTWISE. *Exclude cases with missing values listwise.* A case with a missing value for any variable specified on either `GROUPS` or `VARIABLES` is excluded from any independent-samples test. A case with a missing value for any variable specified on `PAIRS` is excluded from any paired-samples test.

INCLUDE. *Include user-missing values.* User-missing values are treated as valid values.

WEIGHTED KAPPA

`WEIGHTED KAPPA` is available in Statistics Base Edition.

`WEIGHTED KAPPA` is an important generalization of the kappa statistic that measures the agreement of two ordinal subjects with identical categories.

Syntax chart

```
WEIGHTED KAPPA {varlist} {varlist1 BY varlist2}
     [ /MISSING
          [ CLASSMISSING ={TABLE**} {INCLUDE}] ]
     [ /CRITERIA
          [ ASYMPTOTICCILEVEL ={95**} {value}]
          [ WEIGHTING ={LINEAR**} {QUADRATIC}]
          [ IGNORECASE ={FALSE**} {TRUE}] ]
     [ /PRINT
          [ TABULATION({ALL} {varlist1_subset BY varlist2_subset}) ]
          [ CASE({UPPER**} {LOWER}) ]
          [ ORDER({ASCENDING**} {DESCENDING}) ] ]
```

**Default if the subcommand or keyword is omitted.

This command reads the active dataset and causes execution of any pending commands. See the topic Command Order for more information.

Release History

Release 27.0

Example

```
WEIGHTED KAPPA  {varlist} {varlist1 BY varlist2}
```

Overview (WEIGHTED KAPPA command)

Cohen’s kappa statistic is broadly used in cross-classification as a measure of agreement between two observed raters. It is an appropriate index of agreement when ratings are nominal scales with no order structure. `WEIGHTED KAPPA` is an important generalization of the kappa statistic that measures the agreement of two ordinal subjects with identical categories.

Basic specification

```
WEIGHTED KAPPA  {varlist} {varlist1 BY varlist2}
```

A required command to invoke the WEIGHTED KAPPA procedure. Either one variable list or two variable lists connected by **BY** are allowed. The variables are separated by a blank space or multiple blank spaces. The procedure estimates the weighted kappa for all different variable pairs.

Syntax rules

- If one variable list **`varlist`** is specified, it must contain at least two different variables.
- If two variables lists **`varlist1`** and **`varlist2`** are specified, each list must contain at least one variable to run the procedure.
- If both lists contain only one variable, the variables cannot be the same. A certain variable is allowed to be specified in both **`varlist1`** and **`varlist2`**, but the pair of the same variable is not computed.

Operations

- Unlike most transformations, `WEIGHT` takes effect as soon as it is encountered in the command sequence. Thus, special attention should be paid to its position among commands. See the topic Command Order for more information.
- When multiple table lists are in effect, missing values are handled separately for each table.
- Multiple user-missing values, if applicable, are treated as different categories.
- The variable specified first in **`varlist`** or the variable(s) specified by **`varlist1`** are tabulated as the row.

Example

```
WEIGHTED KAPPA  {varlist} {varlist1 BY varlist2}
```

varlist1 BY varlist2

Either one variable list or two variable lists connected by **BY** are allowed.

Variables are separated by a blank space or multiple blank spaces. If one variable list **`varlist`** is specified, it must contain at least two different variables. The procedure estimates the weighted kappa for all different variable pairs. If two variables lists (**`varlist1`** and **`varlist2`**) are specified, each list must contain at least one variable to run the procedure. If both lists contain only one variable, the variables cannot be the same. Certain variables are allowed to be specified in both **`varlist1`** and **`varlist2`**, but the variable pair will not be computed.

CRITERIA Subcommand

The `CRITERIA` subcommand allows the user to specify the options for the estimation of the Cohen’s weighted kappa statistics.

ASYMPTOTIC_CILEVEL(value)
An optional parameter that specifies the confidence level for the estimation of the asymptotic confidence intervals. Must be a single double value in (`0,100`) with 95 as the default setting.

WEIGHTING(LINEAR or QUADRATIC)
An optional parameter that specifies the weighting scale type (either **`LINEAR`** or **`QUADRATIC`**). The use of **`LINEAR`** weights is the default setting.

IGNORE_CASE(boolean)
An optional keyword that controls whether or not the string variables are case sensitive. **`FALSE`** is the default setting, which indicates that strings are case sensitive. When **`TRUE`** is specified, strings are not case sensitive.

PRINT Subcommand

The `PRINT` subcommand controls the crosstabulation tables.

TABULATION(ALL or varlist1_subset BY varlist2_sublist)
An optional keyword that specifies the rating variables that are used in crosstabulation. The keyword is omitted by default, which suppresses the crosstabulation of any rating variables. When the keyword is specified, it must be followed by either the keyword **`ALL`** or `varlist1_subset BY varlist2_subset`. When **`ALL`** is specified, the crosstabulation tables are printed for all different variable pairs specified by `WEIGHTED KAPPA varlist` or `WEIGHTED KAPPA varlist1 BY varlist2`. The variable specified first in **`varlist`** or the variable(s) specified by **`varlist1`** will be tabulated as the row. When `varlist1_subset BY varlist2_subset` is specified, only those crosstabulation tables containing the pairs in **`varlist1_subset`** and **`varlist2_subset`** are printed. Note that if specified, both **`varlist1_subset`** and **`varlist2_subset`** must contain at least one variable, and must be a subset of **`varlist`**, **`varlist1`**, and **`varlist2`**, respectively. The variable(s) specified by **`varlist1_subset`** are tabulated as the row.

CASE(UPPER or LOWER)
An optional keyword that controls whether the string that rates categories in the crosstabulation tables are displayed in uppercase or lowercase. The keyword takes effect in the presence of string variables and `/CRITERIA IGNORECASE = TRUE`. **`UPPER`** is the default setting,which displays the rating categories in uppercase. When **`LOWER`** is specified, the string categories display in lowercase.

ORDER(ASCENDING or DESCENDING)
An optional keyword that controls whether the rating categories in the crosstabulation tables are displayed in ascending or descending order. **`ASCENDING`** is default the setting. The parameter has no affect when the **`TABULATION`** keyword is omitted.

MISSING Subcommand

The `MISSING` subcommand handles the treatment of missing values.

CLASSMISSING(TABLE or INCLUDE
An optional subcommand that controls how missing values are treated. **`TABLE`** is the default setting, which deletes cases with missing values on a table-by-table basis. When multiple table lists are in effect, missing values are handled separately for each table. When **`INCLUDE`** is specified, user-missing values are included. Multiple user-missing values, if applicable, are treated as different categories.

Chapter 4. Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBMproducts. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_. All rights reserved.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBMproducts. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_. All rights reserved.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Index

A

ABS (function)
  MATRIX command 57
ALL (function)
  MATRIX command 57
ALL (keyword)
  CORRELATIONS command 47
  NPAR TESTS command 101, 114
ALPHA (keyword)
  NPTESTS command 91
ANALYSIS (keyword)
  NPAR TESTS command 114
  NPTESTS command 91
  ONEWAY command 120
  T-TEST command 179
ANY (function)
  MATRIX command 57
arithmetic operators
  in matrix language 53
ARSIN (function)
  MATRIX command 57
ARTAN (function)
  MATRIX command 57
ASCENDING (keyword)
  NPTESTS command 95
ASYMPTOTIC_CILEVEL (keyword)
  WEIGHTED KAPPA command 180

B

bar charts
  in Frequencies 14
BAYES REGRESSION (command)
  CRITERIA subcommand 173
  DESIGN subcommand 174
  iteration criteria 173–175
  OUTFILE subcommand 175
  PLOT subcommand 175
  PREDICT subcommand 174
  PRIOR subcommand 175
  SAVE subcommand 174
BINOMIAL (keyword)
  NPTESTS command 92
BINOMIAL (subcommand)
  NPAR TESTS command 102
Bivariate Correlations
  command additional features 8
  correlation coefficients 7
  missing values 8
  options 8
  significance level 7
  statistics 8
BLOCK (function)
  MATRIX command 57
BONFERRONI (keyword)
  ONEWAY command 118
Bonferroni test 118
BOTH (keyword)
  NONPAR CORR command 86
BREAK (statement)
  MATRIX command 72
BROWNFORSYTHE (keyword)
  ONEWAY command 120
BTUKEY (keyword)
  ONEWAY command 118
BY (keyword)
  NPAR TESTS command 101
  WEIGHTED KAPPA command 180

C

C (keyword)
  ONEWAY command 118
CALL (statement)
  MATRIX command 67
CATEGORIES (keyword)
  NPTESTS command 92
CDF.BERNOULLI (function)
  MATRIX command 62
CDF.BETA (function)
  MATRIX command 62
CDF.BINOM (function)
  MATRIX command 62
CDF.BVNOR (function)
  MATRIX command 62
CDF.CAUCHY (function)
  MATRIX command 62
CDF.CHISQ (function)
  MATRIX command 62
CDF.EXP (function)
  MATRIX command 62
CDF.GAMMA (function)
  MATRIX command 62
CDF.GEOM (function)
  MATRIX command 62
CDF.HALFNRM (function)
  MATRIX command 62
CDF.HYPER (function)
  MATRIX command 62
CDF.IGAUSS (function)
  MATRIX command 62
CDF.LAPLACE (function)
  MATRIX command 62
CDF.LNORMAL (function)
  MATRIX command 62
CDF.LOGISTIC (function)
  MATRIX command 62
CDF.NEGBIN (function)
  MATRIX command 62
CDF.NORMAL (function)
  MATRIX command 62
CDF.PARETO (function)
  MATRIX command 62

CDF.POISSON (function)
    MATRIX command 62
CDF.SMOD (function)
    MATRIX command 62
CDF.SRANGE (function)
    MATRIX command 62
CDF.UNIFORM (function)
    MATRIX command 62
CDF.WEIBULL (function)
    MATRIX command 62
CDFNORM (function)
    MATRIX command 57
charts
    templates 122
chi-square
    Fisher's exact test 10
    for independence 10
    in Crosstabs 10
    likelihood-ratio 10
    linear-by-linear association 10
    Pearson 10
    Yates' correction for continuity 10
CHISQUARE (keyword)
    NPTESTS command 92
CHISQUARE (subcommand)
    NPAR TESTS command 103
CHOL (function)
    MATRIX command 57
CILEVEL (keyword)
    CRITERIA subcommand 123
    NPTESTS command 91, 95
CIN (keyword)
    NPAR TESTS command 114
CLABELS (keyword)
    MATRIX command 69
CLASSMISSING (keyword)
    WEIGHTED KAPPA command 181
CLOPPERPEARSON (keyword)
    NPTESTS command 92
CMAX (function)
    MATRIX command 57
CMIN (function)
    MATRIX command 57
CNAMES (keyword)
    MATRIX command 69
COCHRAN (keyword)
    NPTESTS command 97
COCHRAN (subcommand)
    NPAR TESTS command 104
Cochran's statistic
    in Crosstabs 10
Cohen's kappa
    in Crosstabs 10
Cohen's Weighted Kappa 179, 180
column percentages
    in Crosstabs 11
column proportions statistics
    in Crosstabs 11
COMPARE (keyword)
    NPTESTS command 95, 97
COMPUTE (statement)
    MATRIX command 56
contingency coefficient
    in Crosstabs 10
contingency tables 8
CONTRAST (keyword)
    ES subcommand 123
CONTRAST (subcommand)
    ONEWAY command 118
contrasts
    analysis of variance 117
control variables
    in Crosstabs 9
COORDINATES (keyword)
    WEIGHTED KAPPA command 181
correlation coefficients 45
correlations
    in Bivariate Correlations 7
    in Crosstabs 10
    NONPAR CORR command 84
CORRELATIONS (command)
    limitations 45
    matrix output 47
    MATRIX subcommand 47
    MISSING subcommand 47
    PRINT subcommand 46
    significance tests 46
    STATISTICS subcommand 47
    syntax chart 45
COS (function)
    MATRIX command 57
Cramér's V
    in Crosstabs 10
CRITERIA (subcommand)
    BAYES REGRESSION command 173
    NPTESTS command 91
    ONEWAY command 123
    WEIGHTED KAPPA command 180
Crosstabs
    cell display 11
    clustered bar charts 9
    control variables 9
    formats 12
    layers 9
    statistics 10
    suppressing tables 8
crosstabulation
    in Crosstabs 8
CSSQ (function)
    MATRIX command 57
CSUM (function)
    MATRIX command 57
cumulative distribution functions 62
CUSTOM (keyword)
    NPTESTS command 92
CUTPOINT (keyword)
    NPTESTS command 92

D

d
    in Crosstabs 10
DELETEOBJECT (subcommand)
    OUTPUT MODIFY command 129
DESCENDING (keyword)
    NPTESTS command 95
descriptive statistics
    in Frequencies 13

DESCRIPTIVES (keyword)
  CORRELATIONS command 47
  NPAR TESTS command 114
  ONEWAY command 120
DESIGN (function)
  MATRIX command 57
DESIGN (subcommand)
  BAYES REGRESSION command 174
DET (function)
  MATRIX command 57
DIAG (function)
  MATRIX command 57
DISPLAY (statement)
  MATRIX command 84
distribution functions
  cumulative 62
  inverse 65
  probability density 61
  random variable 66
  tail probability 62
DO IF (statement)
  MATRIX command 70
DUNCAN (keyword)
  ONEWAY command 118
Duncan's multiple range test 118
DUNNETT (keyword)
  ONEWAY command 118
Dunnett's C 118
Dunnett's t test 118
DUNNETTL (keyword)
  ONEWAY command 118
DUNNETTR (keyword)
  ONEWAY command 118

E

EFFECTS (keyword)
  ONEWAY command 120
EIGEN (keyword)
  MATRIX command 67
ELSE (statement)
  MATRIX command 70
ELSE IF (statement)
  MATRIX command 70
END IF (statement)
  MATRIX command 70
END LOOP (statement)
  MATRIX command 71
END MATRIX (command) 49
EOF (function)
  MATRIX command 57
EQUAL (keyword)
  NPTESTS command 92
ES (subcommand)
  ONEWAY command 123
eta
  in Crosstabs 10
EVAL (function)
  MATRIX command 57
EXACT (keyword)
  NPAR TESTS command 114
EXCLUDE (keyword)
  CORRELATIONS command 47
  NPTESTS command 91

EXCLUDE (keyword) *(continued)*
  ONEWAY command 120
EXP (function)
  MATRIX command 57
EXPECTED (keyword)
  NPTESTS command 92
expected count
  in Crosstabs 11
EXPONENTIAL (keyword)
  NPTESTS command 92

F

FACTORS (keyword)
  MATRIX command 83
FIELD (keyword)
  MATRIX command 73
FILE (keyword)
  MATRIX command 73, 80
FIRST (keyword)
  NPTESTS command 92, 97
Fisher's exact test
  in Crosstabs 10
FNAMES (keyword)
  MATRIX command 83
FORMAT (keyword)
  MATRIX command 74
FREGW (keyword)
  ONEWAY command 118
Frequencies
  charts 14
  display order 14
  formats 14
  statistics 13
  suppressing tables 14
FREQUENCIES (keyword)
  NPTESTS command 92
frequency tables
  in Frequencies 12
FRIEDMAN (keyword)
  NPTESTS command 97
FRIEDMAN (subcommand)
  NPAR TESTS command 104
FULL (keyword)
  CORRELATIONS command 46
functions
  cumulative distribution 62
  MATRIX command 57
  MATRIX inverse distribution 65
  probability density 61
  random variable 66
  tail probability 62

G

GABRIEL (keyword)
  ONEWAY command 118
Gabriel's pairwise comparisons test 118
Games and Howell's pairwise comparisons test 118
gamma
  in Crosstabs 10
GET (statement)
  MATRIX command 76

GH (keyword)
  ONEWAY command 118
GINV (function)
  MATRIX command 57
Goodman and Kruskal's gamma
  in Crosstabs 10
Goodman and Kruskal's lambda
  in Crosstabs 10
Goodman and Kruskal's tau
  in Crosstabs 10
GRAPHS (subcommand)
  OUTPUT MODIFY command 139
GROUP (keyword)
  NPTESTS command 92, 95
GROUPS (subcommand)
  T-TEST command 178
GSCH (function)
  MATRIX command 57
GT2 (keyword)
  ONEWAY command 118

H

histograms
  in Frequencies 14
Hochberg's GT2 118
HODGES_LEHMAN (keyword)
  NPTESTS command 95, 97
HOLD (keyword)
  MATRIX command 76
HOMOGENEITY (keyword)
  ONEWAY command 120

I

IDENT (function)
  MATRIX command 57
IDF.BETA (function)
  MATRIX command 65
IDF.CAUCHY (function)
  MATRIX command 65
IDF.CHISQ (function)
  MATRIX command 65
IDF.EXP (function)
  MATRIX command 65
IDF.F (function)
  MATRIX command 65
IDF.GAMMA (function)
  MATRIX command 65
IDF.HALFNRM (function)
  MATRIX command 65
IDF.IGAUSS (function)
  MATRIX command 65
IDF.LAPLACE (function)
  MATRIX command 65
IDF.LNORMAL (function)
  MATRIX command 65
IDF.LOGISTIC (function)
  MATRIX command 65
IDF.NORMAL (function)
  MATRIX command 65
IDF.PARETO (function)
  MATRIX command 65
IDF.SMOD (function)
  MATRIX command 65
IDF.SRANGE (function)
  MATRIX command 65
IDF.T (function)
  MATRIX command 65
IDF.UNIFORM (function)
  MATRIX command 65
IDF.WEIBULL (function)
  MATRIX command 65
IF (subcommand)
  OUTPUT MODIFY command 127
IGNORE_CASE (keyword)
  WEIGHTED KAPPA command 180
IN (keyword)
  ONEWAY command 121
INCLUDE (keyword)
  CORRELATIONS command 47
  NONPAR CORR command 87
  NPAR TESTS command 114
  NPTESTS command 91
  ONEWAY command 120
  T-TEST command 179
increment value
  in matrix loop structures 71
INDEPENDENT (subcommand)
  NPTESTS command 95
independent-samples nonparametric tests
  command syntax 89
INDEXING (subcommand)
  OUTPUT MODIFY command 130
indexing clause
  in matrix loop structures 71
indexing variable
  in matrix loop structures 71
individual test names 120
initial value
  in matrix loop structures 71
INV (function)
  MATRIX command 57

J

JEFFREYS (keyword)
  NPTESTS command 92
JONCKHEERE_TERPSTRA (keyword)
  NPTESTS command 95

K

K-S (subcommand)
  NPAR TESTS command 105, 106
kappa
  in Crosstabs 10
KENDALL (keyword)
  NONPAR CORR command 86
  NPTESTS command 97
KENDALL (subcommand)
  NPAR TESTS command 108
Kendall's tau-b
  in Bivariate Correlations 7
  in Crosstabs 10
Kendall's tau-c

Kendall's tau-c *(continued)*
    in Crosstabs 10
KOLMOGOROV_SMIRNOV (keyword)
    NPTESTS command 92, 95
Kolmogorov-Smirnov Z
    NPAR TESTS command 105–107
KRONEKER (function)
    MATRIX command 57
KRUSKAL_WALLIS (keyword)
    NPTESTS command 95
Kruskal's tau
    in Crosstabs 10
KS_SIM (subcommand)
    NPAR TESTS command 107
kurtosis
    in Frequencies 13

L

lambda
    in Crosstabs 10
layers
    in Crosstabs 9
least significant difference 118
LG10 (function)
    MATRIX command 57
LIKELIHOOD (keyword)
    NPTESTS command 92
likelihood-ratio chi-square
    in Crosstabs 10
linear-by-linear association
    in Crosstabs 10
LIST (keyword)
    NPTESTS command 92, 97
LISTWISE (keyword)
    CORRELATIONS command 47
    NONPAR CORR command 87
    NPAR TESTS command 114
    NPTESTS command 91
    ONEWAY command 120
    T-TEST command 179
LN (function)
    MATRIX command 57
logical operators
    in matrix language 54
LOOP (statement)
    MATRIX command 71
LOWER (keyword)
    CORRELATIONS command 46
LSD (keyword)
    ONEWAY command 118

M

M-W (subcommand)
    NPAR TESTS command 108
macro facility
    with MATRIX command 84
    with matrix language 84
MAGIC (function)
    MATRIX command 57
MAKE (function)
    MATRIX command 57
MANN_WHITNEY (keyword)
    NPTESTS command 95
Mann-Whitney U
    NPAR TESTS command 108
Mantel-Haenszel statistic
    in Crosstabs 10
marginal homogeneity test
    NPAR TESTS command 110
MARGINAL_HOMOGENEITY (keyword)
    NPTESTS command 97
matrices
    covariance 47
MATRIX (command)
    BREAK statement 72
    CALL statement 67
    COMPUTE statement 56
    DISPLAY statement 84
    DO IF statement 70
    ELSE IF statement 70
    ELSE statement 70
    END IF statement 70
    END LOOP statement 71
    GET statement 76
    LOOP statement 71
    MGET statement 80
    MSAVE statement 81
    PRINT statement 67
    READ statement 72
    RELEASE statement 84
    SAVE statement 78
    syntax chart 49
    with macro facility 84
    WRITE statement 74
MATRIX (subcommand)
    CORRELATIONS command 47
    NONPAR CORR command 87
    ONEWAY command 121
MATRIX functions 57
MATRIX inverse distribution functions 65
matrix language
    arithmetic operators 53
    column vector 50
    conformable matrices 52
    constructing a matrix from other matrices 52
    control structures 70
    displaying results 67
    functions 57
    logical operators 54
    main diagonal 50
    matrix notation 51
    reading IBM SPSS Statistics data files 55
    reading text files 72
    relational operators 53
    row vector 50
    saving IBM SPSS Statistics data files 55
    scalar 50
    scalar expansion 53
    string variables 51
    symmetric matrix 50
    transpose 50
    variables 50
    with macro facility 84
maximum
    in Frequencies 13

MC (keyword)
    NPAR TESTS command 114
MC_CILEVEL (keyword)
    NPTESTS command 92
MCNEMAR (keyword)
    NPTESTS command 97
MCNEMAR (subcommand)
    NPAR TESTS command 109
McNemar test
    in Crosstabs 10
MDIAG (function)
    MATRIX command 57
mean
    in Frequencies 13
measures of central tendency
    in Frequencies 13
measures of dispersion
    in Frequencies 13
measures of distribution
    in Frequencies 13
median
    in Frequencies 13
MEDIAN (keyword)
    NPTESTS command 95
MEDIAN (subcommand)
    NPAR TESTS command 109
METHOD (subcommand)
    NPAR TESTS command 114
MGET (statement)
    MATRIX command 80
MH (subcommand)
    NPAR TESTS command 110
MIDPOINT (keyword)
    NPTESTS command 92
minimum
    in Frequencies 13
MISSING (keyword)
    MATRIX command 77
    WEIGHTED KAPPA command 181
MISSING (subcommand)
    CORRELATIONS command 47
    NONPAR CORR command 87
    NPAR TESTS command 114
    NPTESTS command 91
    ONEWAY command 120
    T-TEST command 179
missing values
    in Bivariate Correlations 8
    WEIGHTED KAPPA command 181
MMAX (function)
    MATRIX command 57
MMIN (function)
    MATRIX command 57
MOD (function)
    MATRIX command 57
mode
    in Frequencies 13
MODE (keyword)
    MATRIX command 74
MOSES (keyword)
    NPTESTS command 95
MOSES (subcommand)
    NPAR TESTS command 111
MSAVE (statement)
MSAVE (statement) *(continued)*
    MATRIX command 81
MSSQ (function)
    MATRIX command 57
MSUM (function)
    MATRIX command 57
multiple comparisons
    analysis of variance 118

N

NAMES (keyword)
    MATRIX command 77
NCDF.BETA (function)
    MATRIX command 62
NCDF.CHISQ (function)
    MATRIX command 62
NCDF.F (function)
    MATRIX command 62
NCDF.T (function)
    MATRIX command 62
NCOL (function)
    MATRIX command 57
new features
    version 27 1
NONE (keyword)
    NPTESTS command 95, 97
NONPAR CORR (command)
    limitations 85
    matrix output 85
    MATRIX subcommand 87
    MISSING subcommand 87
    missing values 88
    PRINT subcommand 86
    random sampling 85, 87
    SAMPLE subcommand 87
    significance tests 85, 86
    syntax chart 84
    VARIABLES subcommand 86
    with RECODE command 87
nonparametric correlation
    NONPAR CORR command 84
nonparametric tests
    command syntax 89
NORMAL (function) 66
NORMAL (keyword)
    NPTESTS command 92
NOSIG (keyword)
    CORRELATIONS command 46
    NONPAR CORR command 86
NPAR TESTS (command)
    BINOMIAL subcommand 102
    CHISQUARE subcommand 103
    COCHRAN subcommand 104
    FRIEDMAN subcommand 104
    independent-samples test 101
    K-S subcommand 105, 106
    KENDALL subcommand 108
    KS_SIM subcommand 107
    limitations 101
    M-W subcommand 108
    MCNEMAR subcommand 109
    MEDIAN subcommand 109
    METHOD subcommand 114

NPAR TESTS (command) *(continued)*
    MH subcommand 110
    MISSING subcommand 114
    MOSES subcommand 111
    one-sample test 101
    random sampling 114
    related-samples test 101
    RUNS subcommand 111
    SAMPLE subcommand 114
    SIGN subcommand 112
    STATISTICS subcommand 114
    W-W subcommand 113
    WILCOXON subcommand 113
NPDF.BETA (function) 61
NPDF.CHISQ (function) 61
NPDF.F (function) 61
NPDF.T (function) 61
NPTESTS (command)
    CRITERIA subcommand 91
    INDEPENDENT subcommand 95
    MISSING subcommand 91
    ONESAMPLE subcommand 92
    RELATED subcommand 97
    syntax chart 89
NROW (function)
    MATRIX command 57
NSAMPLES (keyword)
    NPTESTS command 92

O

OBJECTPROPERTIES (subcommand)
    OUTPUT MODIFY command 131
observed count
    in Crosstabs 11
one-sample nonparametric tests
    command syntax 89
ONESAMPLE (subcommand)
    NPTESTS command 92
ONETAIL (keyword)
    CORRELATIONS command 46
    NONPAR CORR command 86
ONEWAY (command)
    analysis design 117
    CONTRAST subcommand 118
    contrasts 117
    CRITERIA subcommand 123
    defining factor ranges 117
    ES subcommand 123
    factor variables 117
    limitations 116
    matrix input 121
    matrix output 121
    MATRIX subcommand 121
    MISSING subcommand 120
    missing values 120, 122
    multiple comparisons 118
    orthogonal polynomials 117
    PLOT MEANS subcommand 120
    POLYNOMIAL subcommand 117
    post hoc tests 118
    RANGES subcommand 120
    statistics 120
    STATISTICS subcommand 120
ONEWAY (command) *(continued)*
    syntax chart 115
    TEMPLATE subcommand 122
ORDER (keyword)
    NPTESTS command 95
orthogonal polynomials
    analysis of variance 117
OUT (keyword)
    CORRELATIONS command 47
    NONPAR CORR command 87
    ONEWAY command 121
OUTFILE (keyword)
    MATRIX command 75, 78, 82
OUTFILE (subcommand)
    BAYES REGRESSION command 175
OUTPUT MODIFY (command)
    DELETEOBJECT subcommand 129
    GRAPHS subcommand 139
    IF subcommand 127
    INDEXING subcommand 130
    NAME keyword 126
    OBJECTPROPERTIES subcommand 131
    REPORT subcommand 140
    SELECT subcommand 126
    TABLE subcommand 132
    TABLECELLS subcommand 135
    TEXTS subcommand 139
OVERALL (keyword)
    ES subcommand 123

P

PAIRED (keyword)
    NPAR TESTS command 101
    T-TEST command 178
paired-samples nonparametric tests
    command syntax 89
PAIRS (subcommand)
    T-TEST command 178
PAIRWISE (keyword)
    CORRELATIONS command 47
    NONPAR CORR command 87
    NPTESTS command 95, 97
PARAMETERS (subcommand)
    POWER MEANS INDEPENDENT command 145
    POWER MEANS ONESAMPLE command 148
    POWER MEANS RELATED command 150
    POWER ONEWAY ANOVA command 142, 143
    POWER PARTIALCORR command 153
    POWER PEARSON ONESAMPLE command 156
    POWER PROPORTIONS INDEPENDENT command 158
    POWER PROPORTIONS ONESAMPLE command 161
    POWER PROPORTIONS RELATED command 164
    POWER SPEARMAN ONESAMPLE command 167
    POWER UNIVARIATE LINEAR command 170
PDF.BERNOULLI (function) 61
PDF.BETA (function) 61
PDF.BINOM (function) 61
PDF.BVNOR (function) 61
PDF.CAUCHY (function) 61
PDF.CHISQ (function) 61
PDF.EXP (function) 61
PDF.F (function) 61
PDF.GAMMA (function) 61

PDF.GEOM (function) 61
PDF.HALFNRM (function) 61
PDF.HYPER (function) 61
PDF.IGAUSS (function) 61
PDF.LAPLACE (function) 61
PDF.LNORMAL (function) 61
PDF.LOGISTIC (function) 61
PDF.NEGBIN (function) 61
PDF.NORMAL (function) 61
PDF.PARETO (function) 61
PDF.POISSON (function) 61
PDF.T (function) 61
PDF.UNIFORM (function) 61
PDF.WEIBULL (function) 61
Pearson chi-square
    in Crosstabs 10
Pearson correlation
    Correlations command 45
    in Bivariate Correlations 7
    in Crosstabs 10
percentages
    in Crosstabs 11
percentiles
    in Frequencies 13
phi
    in Crosstabs 10
pie charts
    in Frequencies 14
PLOT (subcommand)
    BAYES REGRESSION command 175
    POWER MEANS INDEPENDENT command 146
    POWER MEANS ONESAMPLE command 148
    POWER MEANS RELATED command 151
    POWER ONEWAY ANOVA command 143
    POWER PARTIALCORR command 153
    POWER PEARSON ONESAMPLE command 156
    POWER PROPORTIONS INDEPENDENT command 159
    POWER PROPORTIONS ONESAMPLE command 162
    POWER PROPORTIONS RELATED command 165
    POWER SPEARMAN ONESAMPLE command 168
    POWER UNIVARIATE LINEAR command 171
PLOT MEANS (subcommand)
    ONEWAY command 120
POISSON (keyword)
    NPTESTS command 92
POLYNOMIAL (subcommand)
    ONEWAY command 117
post hoc tests
    Bonferroni test 118
    Duncan's multiple comparison procedure 118
    Dunnett's C 118
    Gabriel's pairwise comparisons test 118
    Games and Howell's pairwise comparisons test 118
    Hochberg's GT2 118
    least significant difference 118
    ONEWAY command 118
    Ryan-Einot-Gabriel-Welsch multiple stepdown procedure 118
    Ryan-Einot-Gabriel-Welsch's multiple stepdown procedure 118
    Scheffé test 118
    Sidak's t test 118
    Student-Newman-Keuls 118
    Tamhane's T2 118
post hoc tests *(continued)*
    Tamhane's T3 118
    Tukey's b test 118
    Tukey's honestly significant difference 118
    Waller-Duncan test 118
Power Analysis
    statistics 14
POWER MEANS INDEPENDENT
    command syntax 144
POWER MEANS INDEPENDENT (command)
    PARAMETERS subcommand 145
    PLOT subcommand 146
    syntax chart 144
POWER MEANS INDEPENDENT(command)
    limitations 144
POWER MEANS ONESAMPLE
    command syntax 146
POWER MEANS ONESAMPLE (command)
    PARAMETERS subcommand 148
    PLOT subcommand 148
    syntax chart 146
POWER MEANS ONESAMPLE(command)
    limitations 147
POWER MEANS RELATED
    command syntax 149
POWER MEANS RELATED (command)
    PARAMETERS subcommand 150
    PLOT subcommand 151
    syntax chart 149
POWER MEANS RELATED(command)
    limitations 149
POWER ONEWAY ANOVA
    command syntax 141
POWER ONEWAY ANOVA (command)
    PARAMETERS subcommand 142, 143
    PLOT subcommand 143
    syntax chart 141
POWER ONEWAY ANOVA(command)
    limitations 141
POWER PARTIALCORR
    command syntax 152
POWER PARTIALCORR (command)
    PARAMETERS subcommand 153
    PLOT subcommand 153
    syntax chart 152
POWER PARTIALCORR(command)
    limitations 152
POWER PEARSON ONESAMPLE
    command syntax 154
POWER PEARSON ONESAMPLE (command)
    PARAMETERS subcommand 156
    PLOT subcommand 156
    syntax chart 154
POWER PEARSON ONESAMPLE(command)
    limitations 155
POWER PROPORTIONS INDEPENDENT
    command syntax 157
POWER PROPORTIONS INDEPENDENT (command)
    PARAMETERS subcommand 158
    PLOT subcommand 159
    syntax chart 157
POWER PROPORTIONS INDEPENDENT(command)
    limitations 158
POWER PROPORTIONS ONESAMPLE

POWER PROPORTIONS ONESAMPLE *(continued)*
  command syntax 160
POWER PROPORTIONS ONESAMPLE (command)
  PARAMETERS subcommand 161
  PLOT subcommand 162
  syntax chart 160
POWER PROPORTIONS ONESAMPLE(command)
  limitations 161
POWER PROPORTIONS RELATED
  command syntax 163
POWER PROPORTIONS RELATED (command)
  PARAMETERS subcommand 164
  PLOT subcommand 165
  syntax chart 163
POWER PROPORTIONS RELATED(command)
  limitations 163
POWER SPEARMAN ONESAMPLE
  command syntax 166
POWER SPEARMAN ONESAMPLE (command)
  PARAMETERS subcommand 167
  PLOT subcommand 168
  syntax chart 166
POWER SPEARMAN ONESAMPLE(command)
  limitations 166
POWER UNIVARIATE LINEAR
  command syntax 169
POWER UNIVARIATE LINEAR (command)
  PARAMETERS subcommand 170
  PLOT subcommand 171
  syntax chart 169
POWER UNIVARIATE LINEAR(command)
  limitations 169
PREDICT (subcommand)
  BAYES REGRESSION command 174
PRINT (statement)
  MATRIX command 67
PRINT (subcommand)
  CORRELATIONS command 46
  NONPAR CORR command 86
  WEIGHTED KAPPA command 181
PRIOR (subcommand)
  BAYES REGRESSION command 175
probability density functions 61
PROBIT (function)
  MATRIX command 65

Q

QREGW (keyword)
  ONEWAY command 118
QUANTILE REGRESSION
  command syntax 171
QUANTILE REGRESSION (command)
  limitations 172
  syntax chart 171
quartiles
  in Frequencies 13
QUARTILES (keyword)
  NPAR TESTS command 114

R

r correlation coefficient
r correlation coefficient *(continued)*
  in Bivariate Correlations 7
  in Crosstabs 10
random sample
  in nonparametric tests 114
random variable functions 66
range
  in Frequencies 13
RANGES (subcommand)
  ONEWAY command 120
RANK (function)
  MATRIX command 57
rank correlation coefficient
  in Bivariate Correlations 7
rank-order coefficients
  NONPAR CORR command 84
READ (statement)
  MATRIX command 72
RECODE (command)
  with NONPAR CORR command 87
RELATED (subcommand)
  NPTESTS command 97
relational operators
  in matrix language 53
relative risk
  in Crosstabs 10
RELEASE (statement)
  MATRIX command 84
REPORT (subcommand)
  OUTPUT MODIFY command 140
REREAD (keyword)
  MATRIX command 74
RESHAPE (function)
  MATRIX command 57
residuals
  in Crosstabs 11
rho
  in Bivariate Correlations 7
  in Crosstabs 10
risk
  in Crosstabs 10
RLABELS (keyword)
  MATRIX command 69
RMAX (function)
  MATRIX command 57
RMIN (function)
  MATRIX command 57
RNAMES (keyword)
  MATRIX command 69
RND (function)
  MATRIX command 57
RNKORDER (function)
  MATRIX command 57
row percentages
  in Crosstabs 11
RSSQ (function)
  MATRIX command 57
RSUM (function)
  MATRIX command 57
RUNS (keyword)
  NPTESTS command 92
RUNS (subcommand)
  NPAR TESTS command 111
RV.BERNOULLI (function) 66

RV.BETA (function) 66
RV.BINOM (function) 66
RV.CAUCHY (function) 66
RV.CHISQ (function) 66
RV.EXP (function) 66
RV.F (function) 66
RV.GAMMA (function) 66
RV.GEOM (function) 66
RV.HALFNRM (function) 66
RV.HYPER (function) 66
RV.IGAUSS (function) 66
RV.LAPLACE (function) 66
RV.LNORMAL (function) 66
RV.LOGISTIC (function) 66
RV.NEGBIN (function) 66
RV.NORMAL (function) 66
RV.PARETO (function) 66
RV.POISSON (function) 66
RV.T (function) 66
RV.UNIFORM (function) 66
Ryan-Einot-Gabriel-Welsch multiple stepdown procedure 118

S

SAMPLE (keyword)
    NPTESTS command 92, 95
SAMPLE (subcommand)
    NONPAR CORR command 87
    NPAR TESTS command 114
SAMPLEMEAN (keyword)
    NPTESTS command 92
SAMPLEMEDIAN (keyword)
    NPTESTS command 92
SAMPLES (keyword)
    NPAR TESTS command 114
SAVE (statement)
    MATRIX command 78
SAVE (subcommand)
    BAYES REGRESSION command 174
SCHEFFE (keyword)
    ONEWAY command 118
Scheffé test 118
SCOPE (keyword)
    NPTESTS command 91
SE (keyword)
    WEIGHTED KAPPA command 181
SEED (keyword)
    NPTESTS command 91
SELECT (subcommand)
    OUTPUT MODIFY command 126
SETDIAG (keyword)
    MATRIX command 67
SIDAK (keyword)
    ONEWAY command 118
Sidak's t test 118
SIG (function) 61
SIG (keyword)
    CORRELATIONS command 46
    NONPAR CORR command 86
SIG.CHISQ (function) 61, 62
SIG.F (function) 62
SIGN (keyword)
    NPTESTS command 97
SIGN (subcommand)
    NPAR TESTS command 112
SIMULATION (keyword)
    NPTESTS command 92
SIN (function)
    MATRIX command 57
SIZE (keyword)
    MATRIX command 73
skewness
    in Frequencies 13
SNAMES (keyword)
    MATRIX command 84
SNK (keyword)
    ONEWAY command 118
SOLVE (function)
    MATRIX command 57
Somers' d
    in Crosstabs 10
SPACE (keyword)
    MATRIX command 68
SPEARMAN (keyword)
    NONPAR CORR command 86
Spearman correlation coefficient
    in Bivariate Correlations 7
    in Crosstabs 10
SPLIT (keyword)
    MATRIX command 83
SQRT (function)
    MATRIX command 57
SSCP (function)
    MATRIX command 57
standard deviation
    in Frequencies 13
standard error
    in Frequencies 13
STATISTICS (subcommand)
    CORRELATIONS command 47
    NPAR TESTS command 114
    ONEWAY command 120
STEPWISE (keyword)
    NPTESTS command 95, 97
string variables
    in matrix language 51
STRINGS (keyword)
    MATRIX command 79
Student-Newman-Keuls 118
SUCCESS (keyword)
    NPTESTS command 92, 97
sum
    in Frequencies 13
SVAL (function)
    MATRIX command 57
SVD (keyword)
    MATRIX command 67
SWEEP (function)
    MATRIX command 57
SYSMIS (keyword)
    MATRIX command 78

T

T (function)
    MATRIX command 57
T-TEST (command)

T-TEST (command) *(continued)*
  dependent variables 177
  grouping variables 178
  GROUPS subcommand 178
  independent samples 175–177
  limitations 176
  MISSING subcommand 179
  missing values 179
  one sample 176, 177
  paired samples 175, 176, 178
  PAIRS subcommand 178
  syntax chart 175
  test value 177
  TESTVAL subcommand 177
  variable list 178
  VARIABLES subcommand 177
T2 (keyword)
  ONEWAY command 118
T3 (keyword)
  ONEWAY command 118
TABLE (subcommand)
  OUTPUT MODIFY command 132
TABLECELLS (subcommand)
  OUTPUT MODIFY command 135
tail probability functions 62
Tamhane's T2 118
Tamhane's T3 118
tau-b
  in Crosstabs 10
tau-c
  in Crosstabs 10
TEMPLATE (subcommand)
  ONEWAY command 122
templates
  in charts 122
TEST (keyword)
  NPTESTS command 92, 95, 97
tests for independence
  chi-square 10
TESTVAL (subcommand)
  T-TEST command 177
TESTVALUE (keyword)
  NPTESTS command 92, 95
TEXTS (subcommand)
  OUTPUT MODIFY command 139
TIMER (keyword)
  NPAR TESTS command 114
TITLE (keyword)
  MATRIX command 68
total percentages
  in Crosstabs 11
TRACE (function)
  MATRIX command 57
TRANSPOS (function)
  MATRIX command 57
TRIMOUTLIERS (keyword)
  NPTESTS command 95
TRUNC (function)
  MATRIX command 57
TUKEY (keyword)
  ONEWAY command 118
Tukey's b test 118
Tukey's honestly significant difference 118
TWOTAIL (keyword)
TWOTAIL (keyword) *(continued)*
  CORRELATIONS command 46
  NONPAR CORR command 86
TYPE (keyword)
  MATRIX command 80

U

uncertainty coefficient
  in Crosstabs 10
UNIFORM (function)
  MATRIX command 57
UNIFORM (keyword)
  NPTESTS command 92
USERMISSING (keyword)
  NPTESTS command 91

V

V
  in Crosstabs 10
variables
  in matrix language 50
VARIABLES (keyword)
  MATRIX command 77
VARIABLES (subcommand)
  NONPAR CORR command 86
  T-TEST command 177
variance
  in Frequencies 13

W

W-W (subcommand)
  NPAR TESTS command 113
WALD_WOLFOWITZ (keyword)
  NPTESTS command 95
Wald-Wolfowitz test
  NPAR TESTS command 113
WALLER (keyword)
  ONEWAY command 118
Waller-Duncan t test 118
WEIBULL (function) 66
WEIGHTED KAPPA (command)
  CRITERIA subcommand 180
  MISSING keyword 181
  missing values 181
  PRINT subcommand 181
  syntax chart 179, 180
WEIGHTING (keyword)
  WEIGHTED KAPPA command 180
WELCH (keyword)
  ONEWAY command 120
WILCOXON (keyword)
  NPTESTS command 92, 97
WILCOXON (subcommand)
  NPAR TESTS command 113
WITH (keyword)
  CORRELATIONS command 47
  NONPAR CORR command 86
  NPAR TESTS command 101
  T-TEST command 178
WRITE (statement)

WRITE (statement) *(continued)*
  MATRIX command 74

X

XPROD (keyword)
  CORRELATIONS command 47

Y

Yates' correction for continuity
  in Crosstabs 10