FREQUENCIES

```
FREQUENCIES [VARIABLES]=[varlist] [(min,max)] [varlist...]
                        [ALL    ]

[/FORMAT=[{CONDENSE}] [{NOTABLE }] [NOLABELS]
         [{ONEPAGE }]  [{LIMIT(n)}]
       [{DVALUE}] [DOUBLE] [NEWPAGE]]
        [AFREQ ]
        [DFREQ ]

[/MISSING=INCLUDE]

[/BARCHART=[MINIMUM(n)] [MAXIMUM(n)] [{FREQ(n)    }]]
                                      [{PERCENT(n)}]

[/HISTOGRAM=[MINIMUM(n)] [MAXIMUM(n)] [{FREQ(n)    }]]
                                       [{PERCENT(n)}]

            [{NONORMAL}] [INCREMENT(n)]]
             [NORMAL  ]

[/HBAR=same keywords as HISTOGRAM]

[/NTILES=n]  [/PERCENTILES=value list]

[/STATISTICS=[DEFAULT] [MEAN] [STDDEV] [MINIMUM] [MAXIMUM]
             [SEMEAN] [VARIANCE] [SKEWNESS] [SESKEW] [RANGE] [MODE]
             [KURTOSIS] [SEKURT] [MEDIAN] [SUM] [ALL] [NONE]]
```

Example:

```
FREQUENCIES VARIABLES=XVAR YVAR ZVAR1 TO ZVAR5
/FORMAT=NOTABLE
/STATISTICS=ALL
/HISTOGRAM.
```

Overview

FREQUENCIES produces tables of frequency counts and percentages for the values of individual variables.

Defaults

The default output is a table that displays counts for each value of a variable, the counts percentaged over all cases and over all cases with nonmissing values, and the cumulative percentage over all cases with nonmissing values. Values of numeric variables are ordered from lowest to highest; values of string variables are ordered alphabetically. The number of cases with user-missing and system-missing values are displayed. The table is labeled with variable and value labels if these have been defined. Each page displays as many entire single-spaced tables as will fit.

Tailoring

Display Format. You can suppress all tables or only tables for variables with more than a specified number of categories. You can specify one table per page, a condensed format that fits more values in each table, or double spacing, and you can suppress label printing. Table contents can be ordered by values in descending order, by frequencies in ascending order, or by frequencies in descending order.

Statistical Display. You can display percentiles or ntiles and other optional statistics for each numeric variable. Available statistics include mean, median, mode, standard deviation, variance, skewness, kurtosis, and sum.

Plots. Histograms are available for numeric variables, and bar charts are available for numeric or string variables.

Missing Values. You can include user-missing values in statistical calculations and plots.

Syntax

- FREQUENCIES operates via subcommands.
- The minimum specification is the VARIABLES subcommand and a list of variables.

- Subcommands can be named in any order.
- Subcommands must be separated from each other by a slash.
- Keyword order on subcommands is unimportant.
- VARIABLES names the variables to be tabulated.
- All optional subcommands apply to all variables named on the VARIABLES subcommand.
- Specify subcommands VARIABLES, FORMAT, MISSING, and STATISTICS only once.
- Specify only one of the following subcommands at a time: BARCHART, HISTOGRAM, and HBAR. If you specify any two of these together, HBAR is assumed.
- You can use subcommands PERCENTILES and NTILES more than once. Multiple requests for the same percentiles are consolidated.

Operations

- FREQUENCIES is a procedure and causes the data to be read.
- FREQUENCIES tabulates numeric variables with or without decimal values, short string variables, and the short string portion of long string variables.
- Variables are tabulated in the order that they are mentioned on the VARIABLES subcommand. If a variable is mentioned more than once, it will be tabulated more than once.
- Percentages in tables are displayed with one decimal place. Statistics are displayed with three decimal places.
- If a requested ntile or percentile cannot be calculated, a period (.) is displayed.
- The display always uses narrow format regardless of the width defined on SET.
- The HISTOGRAM subcommand on the SET command controls the character used to draw histograms.

Limitations

- The maximum number of variables on a FREQUENCIES command is the same as the system limit.
- The maximum unique observed values over all variables depends on available workspace and on available labels space (up to 5000 characters for all labels combined).

Example

```
FREQUENCIES VARIABLES=XVAR YVAR ZVAR1 TO ZVAR5
/FORMAT=NOTABLE
/STATISTICS=ALL
/HISTOGRAM.
```

- This example requests FREQUENCIES for XVAR, YVAR, and all variables between and including ZVAR1 and ZVAR5.
- The FORMAT subcommand suppresses the display of frequency tables for all variables named on the VARIABLES subcommand.
- The STATISTICS subcommand requests that all statistics available be displayed for each variable named on the VARIABLES subcommand.
- The HISTOGRAM subcommand requests a histogram for each variable named on the VARIABLES subcommand.

VARIABLES Subcommand

VARIABLES names the variables to be tabulated and is the only required subcommand. The actual keyword VARIABLES may be omitted.

- You can use keyword ALL to refer to all user-defined variables in the active file.

FORMAT Subcommand

By default, FREQUENCIES displays as many single-spaced tables with complete labeling information as fit within the page length. The default table is ordered by ascending value (numeric variables) or in alphabetical order (string variables). Use FORMAT with the keywords listed below to change the defaults.

- If you omit the FORMAT subcommand, the defaults are in effect.
- If you specify FORMAT, only defaults that are explicitly altered with keywords are changed.

Table Formats

CONDENSE *Condensed format.* Displays counts in three columns without value labels and with valid and cumulative percentages rounded to integers. Overrides ONEPAGE.

ONEPAGE *Conditional condensed format.* Uses condensed format for tables that would otherwise require more than one page.

NEWPAGE *Each table starts on a new page.*

NOLABELS *No value labels.*

DOUBLE *Double-space frequency tables.*

Table Order

AFREQ *Sort categories in ascending order of frequency.* Ignored when HISTOGRAM, HBAR, NTILES, or PERCENTILES are requested.

DFREQ *Sort categories in descending order of frequency.* Ignored when HISTOGRAM, HBAR, NTILES, or PERCENTILES are requested.

DVALUE *Sort categories in descending order of values (numeric variables)* or in reverse alphabetical order (string variables). Ignored when HISTOGRAM, HBAR, NTILES, or PERCENTILES are requested.

Table Suppression

LIMIT(n) *No frequency tables with more than* n *categories.* The number of missing and valid cases and requested statistics are displayed for suppressed tables.

NOTABLE *No frequency tables.* The number of missing and valid cases are displayed for suppressed tables. Overrides LIMIT.

BARCHART Subcommand

BARCHART produces a bar chart for each variable named on the VARIABLES subcommand. By default, the horizontal axis for each bar chart is scaled in frequencies and the interval width is determined by the largest frequency count for the variable being plotted. Bar charts are labeled with value labels or with the value if no value label is defined.

- If you omit the BARCHART subcommand, no bar charts are displayed.
- If no keywords are specified on the BARCHART subcommand, the default bar charts are displayed.
- Only defaults you explicitly alter with keywords are changed.

MIN(n) *Lower bound* below which values are not plotted.

MAX(n) *Upper bound* above which values are not plotted.

PERCENT(n) *Horizontal axis scaled in percentages,* where optional *n* is the maximum. With no *n* or a too-small *n*, FREQUENCIES chooses 5, 10, 25, 50, or 100, depending on the largest category. This is the default.

FREQ(n) *Horizontal axis scaled in frequencies,* where optional *n* is the maximum. With no *n* or a too-small *n*, FREQUENCIES chooses 10, 20, 50, 100, 200, 500, 1000, 2000, and so forth, depending on the largest category. This is the default.

Example

```
FREQ VAR=JVAR/ BAR=PER MAX(10).
```

- This command requests a bar chart with values through 10 and a percentage scale.
- This example takes advantage of spelling permitted by three-character truncation of keywords.

HISTOGRAM Subcommand

HISTOGRAM displays a plot for each numeric variable named on the VARIABLES subcommand. If there are no specifications on HISTOGRAM, the horizontal axis of each histogram is scaled in frequencies and the interval width is determined by the largest frequency count of the variable being plotted.

- If you omit the HISTOGRAM subcommand, no histograms are displayed.
- If no keywords are specified, the default histograms are displayed.
- Only defaults you explicitly alter with keywords are changed.

MIN(n) *Lower bound* below which values are not plotted.

MAX(n) *Upper bound* above which values are not plotted.

PERCENT(n) *Horizontal axis scaled in percentages*, where optional *n* is the preferred maximum. With no *n* or a too-small *n*, FREQUENCIES chooses 5, 10, 25, 50, or 100, depending on the largest category.

FREQ(n) *Horizontal axis scaled in frequencies*, where optional *n* is the scale. With no *n* or a too-small *n*, FREQUENCIES chooses 10, 20, 50, 100, 200, 500, 1000, 2000, and so forth, depending on the largest category. This is the default.

INCREMENT(n) *Interval width*, where *n* is the size of the interval. Overrides the default number of intervals on the vertical axis which depends on the system page length. For a variable that ranges from 1 to 100, INCREMENT(2) produces 50 intervals with 2 values each.

NORMAL *Superimpose a normal curve.* Based on all valid values for the variable, including values excluded by MIN and MAX.

Example

```
FREQS VAR=VARZ/HIST=NORMAL INCREMENT(4).
```

- This example requests a histogram with a superimposed normal curve and an interval width of 4.
- This example takes advantage of spelling permitted by three-character truncation of keywords.

HBAR Subcommand

HBAR produces a plot for each numeric and string variable named on the VARIABLES subcommand. For numeric variables, HBAR produces a bar chart if the number of categories fits within the page length (see SET). Otherwise, HBAR produces a histogram. HBAR produces bar charts for short string variables and for the short-string portion of long string variables, regardless of the number of values.

By default, the horizontal axis of each plot is scaled in frequencies and the interval is determined by the largest frequency count. All keyword specifications for HISTOGRAM and BARCHART work with HBAR.

PERCENTILES Subcommand

PERCENTILES displays the value below which the specified percentage of cases falls.

- There are no default percentiles.
- If you omit the PERCENTILES subcommand, no percentiles are displayed.

Example

```
FREQUENCIES VARIABLES=VARZ/ PERCENTILES=10 25 33.3 66.7 75.
```

- This example requests the values for percentiles 10, 25, 33.3, 66.7, and 75 for VARZ.

NTILES Subcommand

NTILES calculates the percentages that divide the distribution into the specified number of categories and displays the values below which the requested percentages of cases fall.

- There are no default ntiles for the NTILES subcommand.
- If you omit the NTILES subcommand, ntiles are not displayed.

Example

```
FREQUENCIES VARIABLES=VARZ/ NTILES=4.
```

- This example requests quartiles (percentiles 25, 50, and 75) for VARZ.

STATISTICS Subcommand

The STATISTICS subcommand controls the display of statistics. By default, cases with missing values are excluded from the calculation of statistics.

- If you use the STATISTICS subcommand without any specifications, it produces the mean, standard deviation, minimum, and maximum (the same as produced by the keyword DEFAULT).
- If you use the STATISTICS subcommand with any specifications, only statistics requested are displayed.
- If you omit the STATISTICS subcommand, no statistics are displayed.

The following may be specified on the STATISTICS subcommand:

MEAN *Mean.*

SEMEAN *Standard error of the mean.*

MEDIAN *Median.* Ignored with AFREQ or DFREQ on the FORMAT subcommand.

MODE *Mode.*

STDDEV *Standard deviation.*

VARIANCE *Variance.*

SKEWNESS *Skewness.*

SESKEW *Standard error of the skewness statistic.*

KURTOSIS *Kurtosis.*

SEKURT *Standard error of the kurtosis statistic.*

RANGE *Range.*

MINIMUM *Minimum.*

MAXIMUM *Maximum.*

SUM *Sum.*

DEFAULT *Mean, standard deviation, minimum, and maximum.*

ALL *Display all available statistics.*

NONE *No statistics.*

Example

```
FREQS VAR=AGE/STATS=DEF MODE.
```

- The keyword DEFAULT on the STATISTICS subcommand produces the mean, standard deviation, minimum, and maximum of AGE.
- The keyword MODE requests the mode of AGE.
- This example takes advantage of spelling permitted by three-character truncation of keywords.

MISSING Subcommand

By default, both user- and system-missing values are labeled as missing in the table but are not included in the valid and cumulative percentages, in the calculation of descriptive statistics, or in bar charts and histograms. Use the MISSING subcommand and keyword to alter the handling of missing values.

INCLUDE *Include cases with user-missing values.* Cases with user-missing values will be included in the statistics and plots.

GET

```
GET [FILE={SPSS.SYS**}] [/DROP=varlist]
          {'filename'}
```

**Default if subcommand is omitted.

Example:

```
GET FILE='NEWDATA.SYS'.
```

Overview

GET reads an SPSS/PC system file produced by the SAVE command. A system file contains data and a dictionary with variable and value labels, missing-value flags, and print formats for each variable on the system file.

Defaults

By default, GET retrieves data and dictionary information for all variables on the system file SPSS.SYS in the current directory.

Tailoring

You can retrieve system files other than SPSS.SYS. You can also retrieve only a subset of variables from a system file.

Syntax

- The minimum specification is simply the command keyword.
- Subcommands are separated by an optional slash.
- You cannot use DATA LIST or IMPORT in an SPSS/PC job with GET.

Operations

- GET is a procedure and causes the data to be read.
- System files are designed to be read by SPSS/PC only and should not be edited.
- The data and dictionary become the SPSS/PC active file.
- A file saved with weighting in effect maintains the values of variable $WEIGHT. For a discussion of turning off weights, see WEIGHT.
- The order of cases in a system file depends on their order at the time the file was saved. The values of $CASENUM are those from the raw data file, prior to any selection (see SELECT IF) or sorting (see SORT).

Example

```
GET.
```

- The GET command reads system file SPSS.SYS from the current directory.

FILE Subcommand

Use the FILE subcommand to specify a system file other than the default SPSS.SYS.

- The only specification on FILE is the name of the file.
- The file specification must be enclosed in apostrophes.
- You can specify files residing in other directories by supplying a fully qualified filename (see Universals: Files).

Example

```
GET FIL='\KLDIR\SALDATA.NOV'.
```

- The system file SALDATA.NOV is retrieved from directory KLDIR.
- This example takes advantage of spelling permitted by three-character truncation of keywords.

DROP Subcommand

Use the DROP subcommand to retrieve only a subset of variables from the system file. DROP specifies a variable or list of variables to be dropped. All variables not named on the DROP subcommand will be included in the active file. The SPSS/PC system file on disk is not affected by specifying DROP on the GET command.

- You can specify variables on the DROP subcommand in any order.
- You can use the TO keyword to specify a group of consecutive variables on the SPSS/PC system file.

Example

```
GET FILE='NEWSUM.SYS'/DROP=DEPT79 TO DEPT81, SALARY81, HIRE.
```

- The system file retrieved is NEWSUM.SYS from the current directory.
- All variables between and including DEPT79 and DEPT81, as well as SALARY81 and HIRE, are excluded from the active file.
- All other variables are retained on the active file.

HELP

```
HELP [ALL] [TOPICS] [NEWS]
     [command] [subcommand] [topic]
```

Example:

```
HELP  DESCRIPTIVES.
```

Overview

HELP (alias ?) provides online help during your PC session. HELP messages describe the function, syntax, and operation of commands. HELP also gives information about files, command order, and subcommands for complex procedures. In addition, the HELP facility includes information on changes to the SPSS/PC system since the publication of this manual.

Syntax

- The minimum specification is simply the command keyword.
- Use HELP ALL or HELP TOPICS to obtain a listing of available HELP topics. To obtain help on any of these topics, specify the topic on HELP.
- Use HELP NEWS to obtain information on changes to the SPSS/PC system since the publication of this manual.
- To get help on specific commands or subcommands, specify the command or subcommand name on HELP.
- You can specify only one topic, command, or subcommand, or one of the keywords ALL, TOPICS, or NEWS, at a time.
- You can request help whenever you are prompted with the command prompt.
- You cannot request help within lines of a multiple-line SPSS/PC command.
- You can use three-character truncation of keywords on all help requests.
- You can request HELP with its alias ?

Operations

- HELP is an operations command and is executed immediately.
- HELP without any specifications produces a description of how to use the HELP command.

Example

```
HELP DESCRIPTIVES.
HEL PRO IF.
HELP DATA LIST FIX.
```

- The first HELP command requests information on procedure DESCRIPTIVES.
- The second HELP requests information on the PROCESS IF command.
- The third HELP requests information on DATA LIST FIXED.
- The second and third HELP requests take advantage of spelling permitted by three-character truncation of keywords.

HILOGLINEAR

```
HILOGLINEAR {varlist}(min,max)[varlist(min,max)...]
            (ALL     )

 [/METHOD={BACKWARD}]

 [/MAXORDER=k]

 [ CRITERIA={CONVERGE({0.25**})] [ITERATE({20**})]
                      {n     }           {n   }

            [P({0.05**})] [MAXSTEPS({10**})] [DEFAULT**]]
               {prob  }             {n   }

 [/CWEIGHT={varname }]
           {(matrix)}

 [/PRINT=[DEFAULT**] [FREQ**] [RESID**] [ESTIM**]
         [ASSOCIATION] [ALL]]

 [/PLOT=[{DEFAULT}] [RESID] [NORMPLOT] [NONE**]]
         {ALL    }

 [ /MISSING={LISTWISE**} [INCLUDE)]
            {DEFAULTT   }

 [/DESIGN=[effectname effectname*effectname . . .]]
 [/DESIGN=...]
```

**Default if subcommand is omitted

Example:

```
HILOGLINEAR AVAR(1,2)BVAR(1,2)CVAR(1,3)DVAR(1,3)
/DESIGN=AVAR*BVAR*CVAR DVAR.
```

Overview

HILOGLINEAR fits hierarchical loglinear models to multidimensional contingency tables using iterative proportional-fitting algorithms. HILOGLINEAR also estimates parameters for saturated models. These techniques are described in Everitt (1977), Bishop, Fienberg, and Holland (1975) and Goodman (1978).

Defaults

By default, HILOGLINEAR estimates a saturated model for all variables in the analysis using default options. The default display includes raw and expected cell counts, raw and standardized residuals, and parameter estimates, A case that has a missing value for any variable in the analysis is omitted.

Tailoring

Design Specification. You can request automatic model selection using backward elimination with the METHOD subcommand. You can also specify any hierarchical design and request multiple designs using the DESIGN subcommand.

Design Control. You can control the criteria used in the iterative proportional-fitting and model-selection routines with the CRITERIA subcommand. You can also limit the order of effects in the model with MAXORDER and specify structural zeros for cells in the tables you analyze with CWEIGHT.

Display and Plots. With the PRINT subcommand, you can limit the display for a design or include partiai associations or tests for order of effects for saturated models. You can request residuals plots or normal probability plots of residuals with the PLOT subcommand.

Missing Values. You can control the handling of user-missing values with the MISSING subcommand.

Syntax

- The minimum specification is a variables list with at least two variables followed by their minimum and maximum values.
- The variables list must be specified first.

- The DESIGN subcommand is optional. If no DESIGN subcommand is specified or DESIGN is not the last subcommand, a default model is estimated.
- The PRINT, PLOT, CRITERIA, MAXORDER, and CWEIGHT subcommands should be placed before the designs to which they apply. Other than this, they can appear in any order.
- You can specify multiple PRINT, PLOT, CRITERIA, MAXORDER, and CWEIGHT subcommands. The last of each type specified is in effect for the following designs.
- The PRINT, PLOT, CRITERIA, MAXORDER and CWEIGHT subcommands remain in effect until they are overridden by new subcommands.
- The METHOD subcommand affects only the next design.
- The MISSING subcommand can be specified only once and can be placed anywhere after the variables list.
- Subcommands must be separated by slashes.

Operations

- HILOGLINEAR is a procedure and causes the data to be read.
- HILOGLINEAR builds a contingency table using all variables on the variables list. The table contains a cell for each possible combination of values within the ranges specified for each of the variables.
- HILOGLINEAR assumes there is a category for every integer value in the range of each variable. Empty categories waste space and can cause computational problems. If there are empty categories, you should use the RECODE command to create consecutive integer values for categories.
- Cases with values outside the range specified for any variable are excluded.
- If the last subcommand is not a DESIGN subcommand, HILOGLINEAR prints a warning and generates the default model. This is the saturated model unless MAXORDER is specified. This model is in addition to any that are explicitly requested.
- Only hierarchical log-linear models can be specified.
- If the model is not saturated (for example, when MAXORDER is less than the number of factors), only a goodness-of-fit and the observed and expected frequencies are given.
- The display uses the WIDTH defined on SET. If the defined width is less than 132, some portions of the display may be deleted.

Limitations

- Maximum 10 factors.
- Maximum 1 variables list and 1 MISSING subcommand.

Example

```
HILOGLINEAR AVAR(1,2) BVAR(1,2) CVAR(1,3) DVAR(1,3)
/DESIGN=AVAR*BVAR*CVAR DVAR.
```

- This example builds a 2 × 2 × 3 × 3 contingency table for analysis.
- The DESIGN subcommand specifies the generating class for a hierarchical model. This model consists of main effects for all four variables, two-way interactions among AVAR, BVAR, and CVAR, and the three-way interaction term AVAR by BVAR by CVAR.

Variables List

The required variables list specifies the variables in the analysis.

- Variables must have integer values. If a variable has a fractional value, the fractional portion is truncated.
- Keyword ALL can be used to refer to all user-defined variables in the active file. If ALL is specified, all variables must have the same range.
- A range must be specified for each variable, with the minimum and maximum values separated by a comma and enclosed in parentheses.

- If the same range applies to several variables, the range can be specified once after the last variable to which it applies.
- The variables list must precede all other subcommands.

METHOD Subcommand

By default, HILOGLINEAR tests the model specified on the DESIGN subcommand (or the default model) and does not perform any model selection. All variables are entered and none are removed. Use the METHOD subcommand to specify automatic model selection using backward elimination for the designs that follow.

- You must specify the keyword BACKWARD on the METHOD subcommand.
- The METHOD subcommand affects only the next design.

BACKWARD *Backward elimination.* Perform backward elimination of terms in the model. All terms are entered. Those that do not meet the P criteria specified the CRITERIA subcommand (or the default P) are removed.

MAXORDER Subcommand

The MAXORDER subcommand controls the maximum order of terms in the model estimated for the following designs. If MAXORDER is specified, HILOGLINEAR will test a model only with terms of that order or less.

- MAXORDER specifies the highest-order term that will be considered from the next design. MAXORDER can thus be used to abbreviate computations for the BACKWARD method.
- If the integer on MAXORDER is less than the number of factors, parameter estimates and measures of partial association are not available. Only goodness-of-fit and the observed and expected frequencies are displayed.

Example

```
HILOGLINEAR VARA VARB VARC(1 2)
/MAXORDER=2/DESIGN=VARA*VARB*VARC
/DESIGN=VARA VARB VARC.
```

- This example builds a 2 × 2 × 2 contingency table for VARA, VARB, and VARC.
- The MAXORDER subcommand restricts the terms in the model specified on the first DESIGN subcommand to two-way interactions or less. MAXORDER has no effect on the second DESIGN subcommand since the design requested considers only main effects.

CRITERIA Subcommand

Use the CRITERIA subcommand to change the values of constants in the iterative proportional-fitting and model-selection routines in the designs that follow the subcommand.

- The default criteria are in effect if the CRITERIA subcommand is omitted (see below).
- You cannot specify the CRITERIA subcommand without any keywords.
- Specify each criteria keyword followed by a criterion value in parentheses.
- Only those criteria specifically altered are changed.
- You can specify more than one keyword on CRITERIA.

The following criteria may be specified:

CONVERGE(n) *Convergence criterion.* The default is .25. Iterations stop when the change in fitted frequencies is less than the specified value.

ITERATE(n) *Maximum number of iterations.* The default is 20.

P(prob) *Probability of chi-square for model.* P is in effect only when method BACKWARD is specified. The default is .05.

MAXSTEPS(n) *Maximum number of steps.* MAXSTEPS is in effect only when method BACKWARD is specified. The default is 10.

DEFAULT *Default criteria.* Use DEFAULT to restore defaults changed by a previous CRITERIA subcommand.

CWEIGHT Subcommand

The CWEIGHT subcommand specifies cell weights for a model. The CWEIGHT subcommand is typically used to specify structural zeros in the table.

- You can specify the name of a numeric variable whose values are cell weights or provide a matrix of cell weights enclosed in parentheses.
- You must specify a weight for every cell in the contingency table, where the number of cells equals the product of the number of values of all variables.
- Cell weights are indexed by the values of the variables in the order in which they are specified in the variables list. The index values of the rightmost variable change the most quickly.
- You can use the notation *n***cw* to indicate that cell weight *cw* is repeated *n* times in the matrix.
- CWEIGHT does not weight aggregated input data (see WEIGHT).
- CWEIGHT is ignored with saturated models.

Example

```
HILOGLINEAR AVAR(1,2) BVAR(1,2) CVAR(1,3)
/CWEIGHT=CELLWGT
/DESIGN AVAR*BVAR BVAR*CVAR AVAR*CVAR.
```

- This example weights a cell by the value of the variable CELLWGT when a case containing the frequency for that cell is read.

Example

```
HILOGLINEAR DVAR(1,3) EVAR(1,3)
/CWEIGHT=(0 1 1  1 0 1  1 1 0)
/DESIGN=DVAR, EVAR.

HILOGLINEAR DVAR(1,3) EVAR(1,3)
/CWEIGHT=(0 3*1 0 3*1 0)
/DESIGN=DVAR,EVAR.
```

- These two equivalent HILOGLINEAR commands set the diagonal cells in the model to structural zeros. This type of model is known as a quasi-independence model.
- Because both DVAR and EVAR have three values, weights must be specified for nine cells.
- The first HILOGLINEAR command specifies cell weights explicitly.
- The second HILOGLINEAR command uses the *n***cw* notation to repeat the cell weight. The first cell weight is applied to the cell in which DVAR is 1 and EVAR is 1; the second weight is applied to the cell in which DVAR is 1 and EVAR is 2, and so forth.

Example

```
TITLE AMERICAN BLADDER NUT SEPARABILITY  HARRIS(1910).
SUBTITLE AN INCOMPLETE RECTANGULAR TABLE.
DATA LIST FREE / LOCULAR RADIAL FREQ.
WEIGHT BY FREQ.
BEGIN DATA.
1 1 462
1 2 130
1 3 2
1 4 1
2 1 103
2 2 35
2 3 1
2 4 0
3 5 614
3 6 138
3 7 21
3 8 14
3 9 1
4 5 443
4 6 95
4 7 22
4 8 8
4 9 5
END DATA.
HILOGLINEAR LOCULAR (1,4) RADIAL (1,9)
/CWEIGHT=(4*1 5*0   4*1 5*0   4*0 5*1   4*0 5*1)
/PRINT=ALL
/PLOT=DEFAULT
/DESIGN LOCULAR RADIAL.
```

- This example uses aggregated table data as input.
- The DATA LIST command defines three variables. The values of LOCULAR and RADIAL index the levels of those variables, so that each case defines a cell in the table. The values of FREQ are the cell frequencies.
- The WEIGHT command weights each case (cell) by the value of the variable FREQ. Since each case represents a cell, the cell frequency will have the value of FREQ for that case.
- The BEGIN DATA and END DATA commands enclose the lines of data.
- The HILOGLINEAR variables list specifies two variables. LOCULAR has values 1, 2, 3, and 4. RADIAL has integer values 1 through 4.
- The CWEIGHT subcommand identifies a block rectangular pattern of cells that are logically empty. There is one weight specified for each cell of the 36-cell table.
- PRINT requests all display statistics and PLOT requests the default plot display.
- The DESIGN subcommand specifies only main effects only for LOCULAR and RADIAL. Lack of fit for this model indicates an interaction of the two variables.

PRINT Subcommand

The PRINT subcommand controls the display produced for the following designs.

- If the PRINT subcommand is omitted or included with no specifications, the default display is produced.
- If any keywords are specified on PRINT, only output specifically requested is displayed.

The following may be specified on PRINT:

FREQ *Frequencies.* Display observed and expected cell frequencies. If the defined width is wide enough, HILOGLINEAR also prints cell percentages.

RESID *Residuals.* Display raw and standardized residuals.

ESTIM *Parameter estimates for a saturated model.* ESTIM is included in the default display for saturated models. ESTIM is not available for unsaturated models (including when MAXORDER is less than the number of factors).

ASSOCIATION *Partial associations of effects for a saturated model.* ASSOCIATION is not available for unsaturated models (including when MAXORDER is less than the number of factors). This option is computationally expensive for tables with many factors.

DEFAULT *Default display.* Includes FREQ and RESID for all models and ESTIM for saturated models. This is the default if PRINT is omitted or included without any specifications.

ALL *All available displays.*

PLOT Subcommand

Use the optional PLOT subcommand to request residuals plots for the following designs.

- No plots are displayed for saturated models.
- If the PLOT subcommand is omitted, no plots are produced.
- If PLOT is included without specifications, standardized residuals and normal probability plots are produced.

RESID *Standardized residuals by observed and expected counts.*

NORMPLOT *Normal probability plots of adjusted residuals.*

NONE *No plots.* Specify NONE to suppress plots requested on a previous PLOT subcommand. This is the default if the PLOT subcommand is omitted.

DEFAULT *Default plots.* Includes RESID and NORMPLOT. This is the default when PLOT is specified without keywords.

ALL *All available plots.*

MISSING Subcommand

By default, a case with missing values for any variable named on the variables list is omitted from the analysis. Use the MISSING subcommand to change the treatment of cases with user-missing values.

- The MISSING subcommand can be named only once and can be placed anywhere following the variables list.
- The MISSING subcommand cannot be used without specifications.
- A case with a system-missing value for any variable named on the variables list is always excluded from the analysis.

The following specifications are available for MISSING:

LISTWISE *Delete cases with missing values listwise.* This is the default if the subcommand is omitted. You can also request listwise deletion with keyword DEFAULT.

INCLUDE *Include user-missing values as valid.* Only cases with system-missing values are deleted.

DESIGN Subcommand

The default model is a saturated model that includes all variables in the variables list. A saturated model contains all main effects and interactions for those variables. Use the DESIGN subcommand to specify a different generating class for the model. In a hierarchical model, higher-order interaction effects imply lower-order interaction and main effects. The highest-order effects to be estimated are the generating class.

- If the DESIGN subcommand is omitted or included without specifications, the default model is estimated.
- To specify a design, list the the highest-order terms, using variable names and asterisks (*) to indicate interaction effects.
- Higher-order interaction terms specified on DESIGN imply all lower-order interaction and main effect terms. AVAR*BVAR*CVAR implies the three-way interaction AVAR by BVAR by CVAR, two-way interactions AVAR by BVAR, AVAR by CVAR, and BVAR by CVAR, and main effects for AVAR, BVAR, and CVAR.
- You can specify more than one DESIGN subcommand. One model is estimated for each.
- If the last subcommand on HILOGLINEAR is not DESIGN, the default model will be estimated in addition to models explictly requested.
- Any PRINT, PLOT, CRITERIA, METHOD, and MAXORDER subcommands that apply to a DESIGN subcommand must appear before it.
- All variables named on the DESIGN subcommand must be named or implied on the variables list.

References

Bishop, Y., S. Fienberg, P. Holland. *Discrete multivariate analysis: Theory and practice.* Cambridge: MIT Press, 1975.

Everitt, B. S. *The analysis of contingency tables.* New York: Halsted Press, 1977.

Goodman, L. A. *Analyzing qualitative/categorical data.* Cambridge: Abt Books, 1978.

IF

```
IF (logical expression) target variable=assignment expression
```

Relational Operators:

EQ or =	Equal to	NE or ~= or <>	Not equal to
LT or <	Less than	LE or <=	Less than or equal to
GT or >	Greater than	GE or >=	Greater than or equal to

Logical Operators:

AND or & OR or | NOT or ~

Missing-Value Functions:

SYSMIS	Returns 1 (true) if value is system-missing
MISSING	Returns 1 (true) if value is system- or user-missing
VALUE	Includes all values except system-missing

Example:

```
IF (SEX EQ 'F')EEOVAR=QUOTA+SXVAR.
IF (SYSMIS(QVAR))RVAR=0.
IF (ABS(A-C) LT 100)INT=100.
```

Overview

The IF command conditionally executes a single COMPUTE-like transformation based upon logical conditions found in the data. The transformation can create a new variable or modify the values of an existing variable for each case in your active file. You can create or modify the values of numeric and short string variables.

The IF command has three components. An expression enclosed in parentheses sets up the logical criteria and is called the *logical expression* (see Universals: Logical Expressions). The *target variable* (the one to be modified or created) is named next. Following the target variable is an equals sign and the *assignment expression*. The target variable's values are modified according to the assignment expression.

Syntax

- The minimum specification is a logical expression, followed by a target variable, an equals sign, and an assignment expression.
- The equals sign is required.
- Parentheses around the logical expression are required.
- Parentheses may also be used to specify the order of operations.
- The logical expression can contain string variables, numeric variables, or both.
- At least one relation, SYSMIS function, or MISSING function must be included in the logical expression.
- A relation includes a variable name, the relational operator, and a value or variable.
- String values used in relations must be specified in quotes and must include any leading or trailing blanks.
- A relation cannot compare a string variable to a numeric value or variable, or vice versa.
- Both the logical expression and the assignment expression can use arithmetic operations and functions allowed in COMPUTE transformations (see COMPUTE and Universals: Functions).

Operations

- IF is a transformation and is executed when the data are read for the next procedure.
- Each IF command is evaluated independently.
- The logical expression is evaluated as true or false. The assignment is executed only if the logical expression is true.
- If the logical expression is false or if one of the variables used in the logical expression is system- or user-missing, the assignment is not made. Existing

target variables remain unchanged; new numeric variables are assigned the system-missing value and new string variables are set to blanks.

- Logical expressions are evaluated in the following order: first numeric functions, then exponentiation, then arithmetic operations, then relations, and finally logical operators. You can change the order of operations using parentheses.
- For assignment expressions, the order of evaluation is numeric functions, then exponentiation, and then arithmetic operators.
- Relational and logical operators cannot be used in assignment expressions.
- Numeric variables created with IF are initially set to the system-missing value. Short string variables created with IF are initially set to a blank value of the specified width.
- Numeric variables created with IF are assigned a print format of eight characters with two decimal places.
- String variables created with IF are assigned a width equal to the number of characters used in the initial assignment.

Limitations

- The number of variables created with IF, COMPUTE, and COUNT plus the number defined on DATA LIST, IMPORT, or GET cannot exceed the system maximum of 200 variables.

Example

```
IF (XVAR EQ 5) YVAR=3.
```

- Numeric variable YVAR is set to 3 for cases where XVAR equals 5 (the expression is true).
- When the expression is false or missing, the value of YVAR remains unchanged. If YVAR has not been previously defined, it contains the system-missing value.

Example

```
IF (SEX EQ 'F') EEOVAR=QUOTA+SVAR.
```

- The logical expression tests string variable SEX for the value "F."
- When the expression is true (when SEX equals F), the value of numeric variable EEOVAR is assigned the value of QUOTA plus SVAR. Both QUOTA and SVAR must be previously defined numeric variables.
- When the expression is false or missing, the value of EEOVAR remains unchanged. If EEOVAR has not been previously defined, it contains the system-missing value.

Example

```
COMPUTE SVAR=0.
IF ((QVAR-RVAR) LE 7)SVAR=QVAR**2.
IF (ABS(A-C) LT 100)INT=100.
```

- COMPUTE assigns SVAR the value 0.
- The logical expression tests whether QVAR minus RVAR is less than or equal to 7. If it is, the value of SVAR is assigned the value of QVAR squared. Otherwise, the value of SVAR remains at 0.
- The second IF command tests whether the absolute value of variable A minus variable C is less than 100. If it is, INT is assigned the value 100. Otherwise, the value is unchanged or, if INT has not been previously defined, system-missing.

Example

```
IF (SYSMIS(QVAR))RVAR=0.
COM VALID=0.
IF (NOT(SYSMIS(VARA)))VALID=1.
```

- The first IF command tests whether QVAR is system-missing. If it is, RVAR is assigned the value 0. Otherwise RVAR is unchanged, or system-missing if RVAR has not been previously defined.
- COMPUTE assigns variable VALID a value of 0.

- The next IF command tests whether VARA is not system-missing. For each case where VARA contains a vaild value (is not system-missing), the value of VALID is set to 1. For each case that contains a system-missing value for VARA, the value of VALID equals 0.
- The example takes advantage of spelling permitted by three-character truncation of keywords.

Example

```
IF (STATE EQ 'IL' AND CITY EQ 13)COST=COST + .07 * COST.
```

- The logical expression tests whether STATE equals IL and CITY equals 13.
- If the logical expression is true, numeric variable COST is assigned the original value of COST plus 7% of the original value of COST (1.07 × COST).
- For any other value of STATE or CITY, the value of COST remains unchanged.

Example

```
IF (VALUE(VARA) GT 0)QVAR=AVAR*BVAR.
```

- The logical expression tests whether the value of VARA, including the user-missing value, is greater than 0. If it is, QVAR is assigned the value of AVAR times BVAR.
- For values less than or equal to 0, QVAR remains unchanged.

Example

```
IF (QVAR EQ 'ok')AVAR='fine'.
```

- The new string variable AVAR is set to "fine" when the value of QVAR is "ok."
- AVAR has format width of four characters.
- When QVAR is not equal to "ok," AVAR is defined as a four-column blank field.

Example

```
IF (RECV GT DUE OR (REVNUES GE EXPNS AND BALNCE GT
0))STATUS='SOLVENT'
```

- The IF command specifies a complex logical expression.
- First, SPSS/PC tests whether REVNUES is greater than or equal to EXPNS and whether BALNCE is greater than 0.
- Second, SPSS/PC evaluates if RECV is greater than DUE.
- If either of these expressions is true, STATUS is assigned the value "SOLVENT."
- If both expressions are false, STATUS remains unchanged. If STATUS has not been previously defined, it is defined as a seven-column blank field.

IMPORT

```
IMPORT FILE='filename' [/KEEP={ALL     }] [/DROP=varlist]
                              {varlist}

    [/RENAME=(old varlist=new varlist)...] [/MAP]
```

Example:

```
IMPORT FILE='NEWDATA.POR'/RENAME=(ID,SEX,AGE=V1 TO V3)/MAP.
```

Overview

IMPORT reads a portable ASCII data file and dictionary produced by the SPSS/PC or SPSS^X EXPORT command. You can download files from SPSS^X on a mainframe to your IBM PC using KERMIT, provided KERMIT is installed on both the mainframe and the PC (see Communications, Part F).

Defaults

IMPORT reads all data and dictionary information for all cases on the portable file. The data dictionary contains the variable and value labels, missing-value flags, and print formats for each variable on the portable file. The file also contains a message with the name, release, and version of the originating software, and the date and time the portable file was created. When the file originates from SPSS^X, it also includes the file label and the name of the originating installation. SPSS/PC assumes that the file is in the current directory.

Tailoring

You can import a subset of variables from the portable file, and you can rename the imported variables. You can also obtain a listing of the imported variables. If you have multiple floppy disk drives, you can read a portable file from a floppy disk.

Syntax

- The minimum specification on IMPORT is a file specification.
- Subcommands can be named in any order and must be separated by a slash.

Operations

- IMPORT is a procedure and causes the data to be read.
- A portable file cannot be read with most IBM/PC editors.
- The portable data file and dictionary become the SPSS/PC active file. You cannot specify DATA LIST or GET in an SPSS/PC session that uses EXPORT.
- You cannot read a portable file directly from a directory other than your current directory or from a disk drive used by the SPSS/PC security system.
- The SPSS/PC active file has a more restrictive dictionary than mainframe versions of SPSS^X. SPSS/PC IMPORT changes the dictionary to conform to internal conventions.
- A file saved with weighting in effect automatically uses the case weights when the file is read.

Limitations

- You may not have enough available memory on your PC to read a portable file produced by SPSS^X on a mainframe. When you produce portable files on a mainframe for use on a PC, you should include only the variables you need for particular SPSS/PC jobs.
- You can import a file with up to 200 variables. Each eight-character portion of a long string variable counts as one variable.

Example

```
IMPORT FILE='NEWDATA.POR'/RENAME=(ID,SEX,AGE=V1 TO V3)/MAP.
```

- The portable file is read from NEWDATA.POR in the current directory.
- Variables ID, SEX, and AGE are renamed V1, V2, and V3 in the active file. Their names are unchanged on the portable file on NEWDATA.POR. None of the other variables read into the SPSS/PC active file are renamed.
- The MAP subcommand requests a listing of variables on the active file.

FILE Subcommand

The FILE subcommand specifies the name of the file that contains the portable file.

- The filename must be enclosed in apostrophes.

- By default, the file is located in the current directory.
- You can use a portable file stored on a floppy disk, provided you can maintain the SPSS/PC security system on one drive and the portable file in another drive.

Example

```
IMP FIL='SALDATA.NOV'.
```

- The portable file is read from SALDATA.NOV in the current directory.
- This example takes advantage of spelling permitted by three-character truncation of keywords.

DROP and KEEP Subcommands

Use the DROP and KEEP subcommands to import only a subset of variables from the portable file.

- DROP excludes a variable or list of variables from the active file. All variables not named on DROP are included.
- KEEP names a variable or list of variables to be included in the active file. All variables not named are excluded.
- Variables can be specified on DROP or KEEP in any order.
- With the DROP subcommand, the order of variables remaining on the active file is the same as their order on the portable file.
- With KEEP, the order of variables on the active file is the order in which they are named on KEEP. Thus, you can use KEEP to reorder variables.
- You can use both DROP and KEEP on the same IMPORT command, provided they do not name any of the same variables.
- You can use the TO keyword to specify consecutive variables on the portable file.
- The portable file is not affected by specifying DROP or KEEP on the IMPORT command.

Example

```
IMPORT FILE='NEWSUM.EXP'/DROP=DEPT79 TO DEPT81.
```

- The portable file is read from NEWSUM.EXP in the current directory.
- Variables between and including DEPT79 and DEPT81 are excluded from the active file. All other variables are on the active file.

RENAME Subcommand

Use the RENAME subcommand to rename variables being read from the portable file. The renamed variables retain the variable and value labels, missing-value flags, and print formats assigned in the job where the portable file was created.

- To rename a variable, specify the name of the variable in the active file, an equals sign, and the new name.
- The equals sign is required.
- You can specify lists of variables on both sides of the equals sign. The number of variables on both sides must be the same, and the entire specification must be enclosed in parentheses.
- You can use the TO convention for both variable lists (see Universals: Variable-Naming Conventions).

Example

```
IMPORT FILE='NEWSUM.POR'/DROP=DEPT79 TO DEPT81
    /RENAME=(DEPT82,SALARY82=DEPT,SALARY).
```

- The RENAME subcommand renames DEPT82 and SALARY82 to DEPT and SALARY.
- DEPT and SALARY retain the variable and value labels, missing-value flags, and print formats assigned to DEPT82 and SALARY82.

MAP Subcommand

If you use the RENAME, DROP, or KEEP subcommands to tailor your file, you may find it helpful to produce a listing of your changes with the MAP subcommand.

- The MAP subcommand can be specified as often as you wish.
- The MAP subcommand produces a listing of all actions taken up to that point.
- When the MAP subcommand is last, it produces a listing of the contents of the active file.

Example

```
IMPORT FILE='NEWSUM.POR'/DROP=DEPT79 TO DEPT81/MAP
    /RENAME DEPT82=DEPT SALARY82=SALARY/MAP.
```

- The first MAP subcommand produces a listing of the variables after DROP has excluded the specified variables.
- The RENAME subcommand renames DEPT82 and SALARY82.
- The second MAP subcommand shows the variables after renaming. Since this is the last subcommand specified, it displays the variables that are contained in the active file.

SPSS^X Portable Files

SPSS^X EXPORT command writes a portable file that is read by a number of different mainframe computers as well SPSS/PC. SPSS PC uses a more restrictive dictionary than SPSS^X. If you know that you are creating a portable file that will be read by SPSS/PC, keep the following in mind:

- After reading 200 variables, SPSS/PC stops processing variables. You can avoid this problem by dropping variables when you create the portable file, so that the portable file contains 200 or fewer variables.
- SPSS/PC only allows one missing-value flag per variable and recodes multiple missing-value declarations to the first missing-value indicator. You can override missing-value declarations prior to creating the portable file with SPSS^X.
- SPSS/PC reserves a fixed amount of memory for variable and value labels that is about 5000 characters. After filling up this area, SPSS/PC ignores additional labels.
- SPSS/PC prints values in one of the following formats: DOLLARw.d, COMMAw.d, Aw, or Fw.d. SPSS^X offers additional formats, which SPSS/PC attempts to translate. If the translation doesn't meet your purposes, you can change the print format of variables with the FORMAT command.
- SPSS^X portable system files can also be produced from SAS (Statistical Analysis System) data sets with the SAS procedure TOSPSS. SPSS Inc. distributes TOSPSS to SPSS^X sites and will, on request, provide TOSPSS along with Kermit to SAS sites where SPSS^X is not installed.

INCLUDE

```
INCLUDE 'filename'
```

Example:

```
INCLUDE '\MASTER\SET.INC'.
INCLUDE 'DEFINE.INC'.
INCLUDE 'DOIT.INC'.
```

Overview

The INCLUDE command allows you to execute SPSS/PC commands from a file. With INCLUDE, you can prepare an entire session with your editor and leave SPSS/PC to execute unattended, as in batch-type processing. You can also use INCLUDE to execute all the file definition commands (such as DATA LIST and labeling commands) and then execute analysis commands interactively. Another use for INCLUDE is to execute a "profile" for your machine configuration. For example, you can define printer characters, a prompt, and page size on the SET command and then include that command in any SPSS/PC session. In addition, you can nest INCLUDE commands so that one set of included commands includes another set of commands. This "nesting" can go five levels deep.

Syntax

- The only specification for INCLUDE is a filename enclosed in single quotes.
- You can include a file from another directory by using a fully qualified filename.
- A command file may contain more than one INCLUDE command, either in a series or nested.
- If you INCLUDE a file of inline data, the first line of the data file must contain the BEGIN DATA command. The END DATA command can be specified as the last line of the included file or with your SPSS/PC commands.

Operations

- INCLUDE is an operation command and is executed immediately.
- INCLUDE identifies a file containing SPSS/PC commands or inline data.
- By default, each command from the INCLUDE file is displayed on your screen as it is processed. You can suppress this display by using the INCLUDE OFF specification on SET.
- Both the INCLUDE command and commands from the included file are copied to the LOG file. The INCLUDE command is executable from the LOG file. The commands from the included file are prefaced with an open bracket (|). These commands are treated as comments and are not executable. Thus, if you use the LOG file in a subsequent session, the INCLUDE command is read from the LOG file and the included commands are read from the original file. The included commands are executed only once.
- If an INCLUDE file contains a FINISH command, the SPSS/PC session ends and you are returned to DOS. No subsequent SPSS/PC commands are processed.

Limitations

- SPSS/PC will process only up to five levels of included files at a time.

Example

```
INCLUDE '\MASTER\SET.INC'.
INCLUDE 'DEFINE.INC'.
INCLUDE 'DOIT.INC'.
FINISH.
```

- The first INCLUDE command processes the commands in file SET.INC in directory \MASTER. SET.INC contains the following:

```
SET DISK=ON/LENGTH=59/BOXST='-|+'/BEEP OFF.
```

- The second INCLUDE command processes commands in file DEFINE.INC in the current directory. DEFINE.INC contains the following:

```
DATA LIST FILE='CURRENT.DAT' / MONTH 1-2 (A) DAY 3-4 TEMP 6-7
     PRESSURE 8-12 (2) WINSPED 13-14.
MISSING VALUE DAY (99)/WINSPED (-1).
INCLUDE 'TRANSFOR.INC'.
```

- DEFINE.INC includes another INCLUDE command, which processes commands in file TRANSFOR.INC from the current directory. TRANSFOR.INC contains some data transformation commands. This is an example of a nested INCLUDE.
- The next INCLUDE command processes commands from file DOIT.INC in the current directory. DOIT.INC contains the following commands:

```
FREQ VAR=ALL/HBAR.
DESC VAR=ALL.
```

- The FINISH command signals the end of the SPSS/PC session.