Short manual for tabulation program

Version 1.6.7

Contents

[Command STABLE 3](#_Toc212874648)

[Defining aliases 18](#_Toc212874649)

[How to use TABOFFREQ command 24](#_Toc212874650)

[How to define matching labels with specific colors 26](#_Toc212874651)

[How to define multiple variables 28](#_Toc212874652)

[How to define block multiple variables 30](#_Toc212874653)

[How to use BIGTAB command 32](#_Toc212874654)

# Command STABLE

## Syntax:

!stable type = [simple | multiple | binary | marks | average | median /

vars = <variable names >/

col1 – col15 = <variable names for crosstabs in frits till fiftinth table*>*/

brkat = <*number of marks, when marks type is used*>/

color = [0 | 1 | 2 | 3]/

decimal = <*two numbers*> | <*one number*>/

ltperc = <*two numbers*>/

srtvl = <*one number*>/

value = <*one number*> | <*more then one number*>/

label = <*label-heading for table*>/

filter = [<name of filter variable for each table> | all]/

select = *numeric expresion*/

chart = <two numbers> | <*one number*>/

missing = [RF | DK | NORF | NODK | <*one number*> | <*more then one number*>] | NONE /

boxes = "Bottom 3 boxes" 1:3 "Top 4 boxes" 7:10 /

options = ascending

descending

nozerocolumns

unwn

non

nobase

samplesize

ownbase

whole

count

share

trim

top5

nototal1

nototal2

nototal3

nototal4

nototal5

transpose

plus

labvals

rlabvals

clabvals

nochart

ascending label\_sort

descending label\_sort

varname

rowpct

sigc

sigw

abc

nobonf

nps/.

## type

This argument defines the type of table which will be produced. In one command only one type can be defined.

* simple designates that for each listed variable in argument vars separate table will be produced
* multiple designates that for all listed variable in argument vars will be treated as one multiple categorical variable, in which every value designates different category
* binary designates that for all listed variable in argument vars will be treated as one multiple binary variable, in which every variable designates different category
* marks similar to simple, with that difference that there will be added two more categories, sum of all “negative” and sum of all “positive” answers. In table there will be row more with averages of values from 1 till the value specified in argument brkat.
* average table of averages of variables specified in vars
* median table of medians of variables specified in vars

## vars

This argument represents the names of variables that we would like to have in top line of table, i.e. the variablems that we would like to tabulate. If type of table is simple and we specify the multiple variables the multiple type of this command will be issued.

## col1 – col5

These five arguments represent the variables that are going to be put in columns of five tables that tabulate each one variable from vars argument. Specified variables can be ordinary simple, categorical multiple or binary multiple variables. First column in each table will be “Total“. In order to omit this column we need to specify the nototal1 option. Option nototal1 is for the first table i.e. nototalX where X represents the number of table for which we would like to omit the “Total”.

The system variable $TOTAL represents the total of all answers from whole sample, and with this variable and option nototal it is possible to move “Total” column from first position to end of table, or somewhere in between.

Specified variables will be presented in table in that order. If we would like to nest variable in previous one, we need to use operator „>“.

See examples 2 i 11

## brkat

Specified number under this argument represents the number of valid answers. This argument makes sence only in marks table type.

Default value for this argument is 5. In that case, rows will be added. First one that will represent the sum of answers 1 i 2, and second one with sum of rows 4 i 5. Row that will represent the average will be calculated just for the answers with values from 1 to 5; all other values will be shown as percentages but will not be included in calculation of average.

## Color

Number of color shades, or more precisely number of significance levels which will be shown in tables.

Valid values are 0, 1, 2 and 3. Default value, if argument is omitted is 2.

0 represent that no coloring will be used. 1 means that only the most significant differences will be colored, significant on the level of 0.01. If the value is 2, differences significant on level of 0.05 will be colored too. And value of 3 means that that significance differences of 0.10 will be colored. Level 3 should be used if the sample is less than 200 cases.

## decimal

This argument consists of one or two numbers. If two numbers are specified, firs one specifies the number fo decimals used in first column, and second one in all other columns of table.

If only one number is specified it represents the number of decimal places used for numbers in all columns of table.

Default value is 1 i 0, i.e. the first column will have one decimal place, and all other will be without decimal numbers.

See examples 6 and 7

## ltperc

This argument consists of two numbers. First one represents the value in percent below which all the categories of varialbe will be recoded into the value specified in second number of argument.

If the value of second number does not have the label in SPSS file it will be shown as number in table.

See example 4

## srtvl

Name of this argumnet is made of „SoRT VaLue“. It represents the value until values will be sorted.

Point of this argument is to make simple of sorting but still to keep special answers like “Do not know”, “Refuse to answer” and “Other” on the end of list (in last rows of table).

Usually this argument is used together with argument ltperc when we recode all categories with frequencies less than 5% into “Other“, which is coded with big number like 95, and then we sort values until 95. By this category “Other” will be on the end of table.

See example 4

## value

In case we are producing the tables with binary variables (binary) we need to specify values that represent positive answer. This argument represents one or more values i.e. categories that will be treated (counted) as positive answers on variable.

If the argument is omitted, its default value is 1.

See examples 8 and 9

## label

This is string that will be used as title for table. In case that argument is ommited variable label will be used as title.

If you want to use the full stop, equals, division or percent sign (. = / %) in label, put the text in double quotes..

See examples 3, 4, 6, 8, 9 and 10

## filter

For each table (defined by col1 to col5 arguments) it is possible to define filter variable – variable on which zero value means that record/respondent will not be included in analysis.

This is the way to produce tables for different subsamples.

If omitted all records (respondents) will be included in analysis, as the parameter all is specified.

IMPORTANT: Base for analysis will be reported according to the last base specified i.e. last filter specified, so it is maybe better not to print it at all (see option nobase). Difference in comparison with command DataSelect is that with this one selected subsample is going to be persistent for all following analysis, while filter is just for specified table, ratio is made according to all cases, not just filtered.

See example 11

## Select

For analysis it is possible to define numeric expression, and if that expression evaluates at 0 the record will be excluded from analysis

If omitted all records (respondents) will be included in analysis.

Valid operators are +, -, \*, and for division as character ‘/’ is used for argument separation use ’./’ instead. For example:

|  |  |
| --- | --- |
| **argument** | **selects all respondents for whom ...** |
| select = a2./2 < a3/ | half of the value on a2 is less then value of variable a3 |
| select = a1 in [2,11]/ | value of a1 is 2 or 11 |
| select = a1==2 | a1==11/ | same as previous |
| select = a1==2 & a2==5/ | value of a1 is 2 AND value of a2 is 5 |
| select = a1 in [(1:5)',8] | value of a1 is 1 to 5 or 8 |
| select = a1 ~= 27 | value of a1 is different from 27 |

## chart

By default, if output with charts is selected, before table chart will be produced. If this argument is omitted, there will be one or two charts produced, depending on strength of association of crosstabs.

This argument can be issued as one or two numbers, these numbers represents the variables shown in the columns of table which we want to be charted. Number 1 represents the first variable in the table (by default this is “total” column), next one is 2, etc. If we specify 0, then the biggest association with the crosstab variable will be selected, if there is any crosstab with Cramer’s V at least of 0.20.

If argument is omitted, default values are 1 and 0, i.e. left chart will represent total column and the right one the biggest association regarding Cramer’s V coefficient if there is any bigger than 0.20. If we state just 1 for this argument then just one chart will be produced with total column (by default it is the first column). If we state 0 then just one chart will be produced for the crosstab with the biggest association, if there is any bigger than 0.20, if not the total column will be produced.

Example:

chart=4 2/ is command that left in left chart presents crosstabs with fourth variable, and right one with the second..

## missing

For each table it is possible to specify up to three values which we want to be excluded from tabulation (in SPSS’s terminology, this values are “user missing” values).

If we specify the NONE then all the values will be valid ones i.e. they are not going to be excluded from current tabulation.

There are four key words: RF, DK, NORF and NODK, but for their use it is needed to previously define RF and/or DK as global variables and to defined their values. In other cases it is possible to specify the values which we want to exclude form analysis.

If we want to exclude some values form more than one analysis, command missing should be used.

Example:

|  |  |
| --- | --- |
| !stable type = multiple/  vars = f1 f2 f3 f4 f5/  col1 = dem1 stratum tip/  missing = 998 999. | In this example for this multiple variable i.e. for all five f variables, values 998 and 999 will be excluded. This exclusion is valid just for this command. |

Example:

|  |  |
| --- | --- |
| global RF, DK  RF = 999;  DK = 998;  !stable type = simple/  vars = a1/  missing = RF NODK. | In front of first stable command global varilabes are defined and their values aswell. This definition is needed to be executed only once in one session of program, it will persist in all further commands. For variable a1, value 999 (RF) will be excluded, and value 998 (DK) will be shown, regardless of definitions in SPSS file, because of option NODK. |

## options

This argument consists of list of options in that can be defined in current analysis. As all arguments it can be defined just once in one command, if there is more than one definition latter one will be taken into the account.

### ascending

This option will sort percentages, values in “total” i.e. the first column in ascending order, from the first to the last row of table.

### descending

This option will sort percentages, values in “total” i.e. the first column in descending order, from the first to the last row of table.

### nozerocolumns

All columns that have subsample size (row noted with “N” ort wit “unw N”) equal to zero will be excluded from table. This option is usefull when we nest variables in coluns because it produces smaller tables with smaler number of columns.

### unwn

This option will produce the row with unweighted sizes of subsamples. If this option is omitted this row will not be shown, too.

See examples 6 and 7

### non

By default for each column weighted size of subsample will be shown, row designed with “N”. If this option is used that row will be omitted.

### nobase

Under every title of table, size of basis that is used for table calculation is shown in percentages. If this option is stated that will be omitted. It is useful when we use filter argument with more than one table, because in that case shown bases is calculated regarding the first table.

See example 5

### samplesize

If we by weight do expansion of sample onto the population, projection of frequencies onto the population, in that case sum of weight is much bigger than sample size, and in order to calculate significances according to sample size, which is correct way to calculate significances, it is needed to state this option.

See example 7

### ownbase

Tables of binary type are by default calculated on that way that for all variables same basis is used. If there is 5 valid answers on the first variable, and 3 on the second one, second one will be calculated regarding larger basis of minimum 5 respondents; so if all respondents on the second variable gave positive answer it will be maximum of 60%. If we want that each variable uses its own basis we need to specify this option and in that case percentage for the second variable, in used example, will be 100% because all three gave positive answer. If this option is stated and if variables have different basis, row with „N“ and row with „unw N“ will be empty.

See example 9

### whole

For tables of binary type this option will show percentages calculated on basis of all respondents no matter that each variable has its own basis.

See example 9a

### count

This option will show frequencies instead of percentages in table.

See example 11

### share

This option will produce column shares instead of default column percentages in cases we use multiple or average table types. It is incorrect to use it in case of simple table type, even dough the percentages are the same, significances are calculated with less precision than in standard procedure. In case we use this option with multiple table type share from all answers is obtained, and in case we use it in case of average table type classical share is obtained (in order to get more robust shares it is advised to use option trim).

See example 10

### trim

This option is important for calculation of mean values in case of average table types or shares in case of share option. By this option more robust estimates of average values are obtained (more robust in case of the outliers). By using this option 5% of highest and 5% of lowest values are excluded and mean value is calculate on the basis f 90% of middle values.

### top5

By default in right chart with most significant crosstabs, shown categories of variable are those that by most have contributed to association of column variable and tabulated variable. This can often mean that in the right chart shown categories are not and most frequent ones, because it is possible that those categories are not at all, the ones that are responsible for association. This option will force charting of the top five of the most frequent categories.

### nototal1 - nototal5

These five options are used to omit the “total” column from the start of the table. This makes sence when our sample is not representative one or when we just compare two different samples, and when the result obtained from the both samples doesn’t make sense any more.

See example 5

### transpose

Transposes table, rows will be shown as columns and columns will be shown in rows.

### plus

This option is changing the labels for “Sum -” and “Sum +” in marks table typ. Without this option label “Sum -“ will mark the sum of the first half of codes with the smallest values, and with this option that label will be “Sum +”.

### labvals

Tabulated values are by default taken from frequencies i.e. if there is no answer with that value it will not be tabulated. This option is forcing that all the values that have defined labels, will be tabulated no matter if they have zero frequencies or not. This will be applied to row and column variables.

### rlabvals

This option is same as labvals, but just for row variables.

### clabvals

This option is same as labvals, but just for column variables.

### nochart

If specified this option will omit production of chart for that analysis.

### ascending label\_sort

This option will sort percentages, values in ascending order according to names of labels, from the first to the last row of table.

### descending label\_sort

This option will sort percentages, values in descending order according to names of labels, from the first to the last row of table.

### varname

This option explains the variable with the table title (Dem2. Age).

### rowpct

This option calculates Row % in tables.

### nps

Adds row with NPS (Net Promoter Score) sum of 9 and 10 minus sum of 0 to 6 marks.

### sigc

This option enables statistically significant differences between groups.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  | **Gender** | | |  | Males | Females | | N | 500 | 500 | | sig | 0.00 | | |  | A | B | | Brand X | 38 | 24 | | B |  | | Brand Y | 29 | 36 | |  | a | | Brand Z | 32 | 39 | |  | a | | Total | 100% | | | * There are labels related to **statistically** **significant differences between groups**. * Each group is marked with a **specific letter** (A, B, C etc.). * Statistically significant differences are marked with specific letters as well. * Letter denoting a specific group is placed below the value that is significantly higher. * **Capital letter** refers to a confidence interval 99% and **small letter** refers to a confidence interval 95%. * For instance, in the example above, males prefer brand X significantly more than females do, while females prefer brands Y and Z significantly more than males do. |

### sigw

This option enables statistically significant differences between waves – last wave is compared with previous.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Total** | **Talas** | | | | | | | | | | | |
|  |  | wave 1 | wave 2 | wave 3 | wave 3 | wave 5 | wave 6 | wave 7 | wave 8 | wave 9 | wave 10 | wave 11 | Wave12 |
| 1N | 26310 | 1005 | 1008 | 1009 | 1009 | 1000 | 1007 | 1031 | 1034 | 1040 | 1048 | 1032 | 1031 |
| sig |  | 0.00 | | | | | | | | | | | |
| Brand X | 08.3 | 09.0 | 10.2 | 08.7 | 08.7 | 07.6 | 09.0 | 07.0 | 05.0 | 03.5 | 05.0 | 06.0 | 03.7 |
| Brand Y | 19.3 | 21.0 | 20.3 | 22.7 | 18.4 | 20.9 | 20.9 | 21.4 | 18.9 | 24.9 | 23.9 | 26.7 | 28.0 |
| Brand Z | 69.9 | 67.3 | 66.8 | 65.1 | 69.9 | 68.0 | 67.4 | 69.6 | 74.0 | 70.5 | 69.1 | 66.8 | 67.6 |
| DK-Ref | 02.5 | 02.7 | 02.7 | 03.4 | 03.0 | 03.4 | 02.6 | 02.0 | 02.1 | 01.0 | 01.9 | 00.5 | 00.7 |
| Total | 100% | | | | | | | | | | | | |

* Statistically significant differences are marked with colours.
* Cells of the table are colored in blue, or red, if values they contain are considerably above or below the values in previous wave. Two shades of blue or red color are used for three degrees of significance, lighter shades for deviations significant on the level 0.05 and the darkest shades of blue and red for the level 0.01.

### abc

All columns within one variable when sigc option is present are named from the A, like ABCABCD, and not ABCDEFG.

### nobonf

Disables Bonferroni correction for multiple-comparisons.

### heading(2,"chapter title") - This option before the sintaxes will show chapter title in heading2 style.

## Examples

### Example 1

|  |  |
| --- | --- |
| Data | Syntax |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **A1** | **A2** | **POL** | **GOD** | | 1 | apple | peach | male | young | | 2 | apple | . | male | young | | 3 | pear | . | female | young | | 4 | pear | . | female | old | | 5 | . | . | female | old |   . | !stable type = simple/  vars = a1/  col1 = gender age . |
| Resulting output | Comments |
|  | * Just 4 respondents are reported, because fifth has only sysmis values (missing values), so in the basis 80% is reported, because whole database has 5 records. * In table column percentages are shown * As variable A1 didn’t had defined label (in SPSS file), title of table is just its name |

### Example 2

|  |  |
| --- | --- |
| Data | Syntax |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **A1** | **A2** | **POL** | **GOD** | | 1 | apple | peach | male | young | | 2 | apple | . | male | young | | 3 | pear | . | female | young | | 4 | pear | . | female | old | | 5 | . | . | female | old |   . | !stable type = multiple/  vars = a1 a2/  col1 = gender > ($total age) .  Or  !multiple name = $a/  vars = a1 a2 .  !stable type = simple/  vars = $a/  col1 = geder > ($total age). |
| Resulting output | Comments |
|  | * All percentages are calculated regarding 4 respondents that have at least one valid data * For each category of variable GENDER categories of variable AGE with total in front are shown (system variable $total in the syntax). |

### Example 3

|  |  |
| --- | --- |
| Data | Syntax |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **A1** | **A2** | **A3** | **POL** | | 1 | apple | apricot | plum | male | | 2 | apple | . | . | male | | 3 | pear | . | . | female | | 4 | pear | . | . | female | | 5 | peach | . | . | female |   . | !stable type = multiple/  vars = a1 a2 a3/  col1 = gender/  options = descending/  label = "Favorite fruit". |
| Resulting output | Comments |
|  | * All 5 respondents are included in calculation. “Pear” and “apple” are reported by 2 respondents each, which is 40%, while “peach”, “apricot” and “plum” is reported by one respondent, which makes 20%. Out of two male respondents one reported 3 different fruits, so the “apricot” and “plum” within male respondents 50%. * Title is printed using the argument label * descending option is used to sort answers in descanting order |

### Example 4

|  |  |
| --- | --- |
| Data | Syntax |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **A1** | **A2** | **A3** | **POL** | | 1 | apple | apricot | plum | male | | 2 | apple | . | . | male | | 3 | pear | . | . | female | | 4 | pear | . | . | female | | 5 | peach | . | . | female |   . | !stable type = multiple/  vars = a1 a2 a3/  col1 = gender/  options = descending/  label = "Favorite fruit (above 20%)"/  ltperc = 21 95/  srtvl = 95. |
| Resulting output | Comments |
|  | * Data is the same as in previous example, but this time argument ltperc is used, with whom all the answers with percentages less than 21 we recoded into the „Other“ (95) – “peach“, “apricot” and “plum” are recoded into the “other“ * This is 3 different answers (“peach”, “apricot” and “plum”) but the percentage is same as for “apple” and “pear” because one respondent reported two answers that are recoded into the “other” which is counted as one answer * By usage of argument srtvl, in descending order answers are sorted up to value of 95, so the answer “other” is on the end of table |

### Example 5

|  |  |
| --- | --- |
| Data | Syntax |
| |  |  |  | | --- | --- | --- | |  | **SCALE** | **GENDER** | | 1 | Very bad | male | | 2 | Very bad | male | | 3 | Very bad | female | | 4 | Bad | female | | 5 | Good | female | | 6 | Very good | male | | 7 | Very good | female | | 8 | 99 | female |   . | !stable type = marks/  vars = scale/  col1 = gender $total/  options = nototal1 nobase. |
| Resulting output | Comments |
|  | * As argument brkat is omitted default value is 5 * With 1 answer “Very bad” is coded, and with 5 “Very good” * In data there is value of 99 which is shown in table with 13% on total, but was excluded form calulation of average which is 2.7 on total. * Column “Total” is on the end of table, because there is option nototal and system variable $total is on the end of the list of variables under argument col1 (total on the end of table is in comparison of standard outpu shown without of decimal numbers) * Basis is not show because there is option nobase |

### Example 6

|  |  |
| --- | --- |
| Data | Syntax |
| .   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | **A1** | **A2** | **GENDER** | **AGE** | **WGH** | | 1 | 1 | 3 | male | young | 10 | | 2 | 1 | . | male | young | 10 | | 3 | 2 | . | female | young | 10 | | 4 | 2 | . | female | old | 10 | | 5 | . | . | female | old | 10 |   . | DataWeight("WGH")  !stable type = average/  vars = a1 a2/  col1 = gender age/  options = descending unwn/  decimal = 1/  label = How much on average. |
| Resulting output | Comments |
|  | * Data is weighted by variable WGH * In all columns values are shown with one decimal place * Because option unwn row with unweighted subsamples sizes is shown * Values are sorted in descending order, and it this example, row A2 is on the first place and A1 after * Title of table is specified in argument label |

### Example 7

|  |  |
| --- | --- |
| Data | Syntax |
| .   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | **A1** | **A2** | **GENDER** | **AGE** | **WGH** | | 1 | 1 | 3 | male | young | 10 | | 2 | 1 | . | male | young | 10 | | 3 | 2 | . | female | young | 10 | | 4 | 2 | . | female | old | 10 | | 5 | . | . | female | old | 10 |   . | DataWeight("WGH")  !stable type = average/  vars = a1 a2/  col1 = gender age/  options = descending unwn samplesize /  decimal = 1/  label = How much on average. |
| Resulting output | Comments |
|  | * Data and syntax is the same as in previous example, with exception that now there is option samplesize, so the significances are calculated on the basis of true size of sample (4 respondents), and not on the basis of weight like in previous example (size was 40). Because of this difference between younger and older respondents on the variable A1 is not significant. * The second and other columns have two decimal digits, and the first column with one |

### Example 8

|  |  |
| --- | --- |
| Data | Syntax |
| .   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **B1** | **B2** | **B3** | **GENDER** | | 1 | Yes | Yes | . | male | | 2 | Yes | No | . | male | | 3 | Maybe | . | . | female | | 4 | No | . | Yes | female | | 5 | . | . | Yes | female |   . | !stable type = binary/  vars = b1 b2 b3/  col1 = gender/  options = descending/  label = Percent of not negative answer/  value = 1 2. |
| Resulting output | Comments |
|  | * Basis is 5 respondents because every respondent has at least on one variable valid answer * Answers that are counted as positive have values of 1 and 2 i.e. „Yes“ i „Maybe“ * The bigest percentage (60%) is obtained on variable B1, because three out (of five) respondents answered with 1 or 2, and the lowest percentage (20%) is obtained on variable B2 because just one respondent did answered with “Yes” (out of 5 respondents) |

### Example 9

|  |  |
| --- | --- |
| Data | Syntax |
| .   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **B1** | **B2** | **B3** | **GENDER** | | 1 | Yes | Yes | . | male | | 2 | Yes | No | . | male | | 3 | Maybe | . | . | female | | 4 | No | . | Yes | female | | 5 | . | . | Yes | female |   . | !stable type = binary/  vars = b1 b2 b3/  col1 = gender/  options = descending ownbase /  label = Percent of not negative answer/  value = 1 2. |
| Resulting output | Comments |
| **Percent of not negative answer**   |  |  |  |  | | --- | --- | --- | --- | |  | **Total** | **GENDER** | | |  |  | Male | Female | | N | 4 | 2 | 2 | | B3 | 100.0 |  | 100 | | B2 | 100.0 | 100 |  | | B1 | 075.0 | 100 | 050 | | * Data is the same like in previous example with modification that this time we have additional option ownbase because of which every variable has its own basis for calculation * Notice that total N is now 4 because it is the biggest number of answers on one variable , on variable B1 * Now B3 has the biggest percentage, 100%, because both respondents that did answered, had answered with “Yes”, and the smallest percentage there is on variable B2, 50%, because from two respondents that did answered, just one gave positive answer |

### Example 9a

|  |  |
| --- | --- |
| Data | Syntax |
| .   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **B1** | **B2** | **B3** | **GENDER** | | 1 | Yes | Yes | . | male | | 2 | Yes | No | . | male | | 3 | Maybe | . | . | female | | 4 | No | . | Yes | female | | 5 | . | . | Yes | female |   . | !stable type = binary/  vars = b1 b2 b3/  col1 = gender/  options = descending whole/  label = Percent of not negative answer/  value = 1 2. |
| Resulting output | Comments |
| **Percent of not negative answer**   |  |  |  |  | | --- | --- | --- | --- | |  | **Total** | **GENDER** | | |  |  | Male | Female | | N | 5 | 2 | 3 | | B1 | 060.0 | 100 | 033 | | B3 | 040.0 |  | 067 | | B2 | 040.0 | 100 |  | | * Data is the same like in previous example with modification that this time we have additional option whole because percetages are calculated o basis of all respodents. * Notice that total N is now 5 – total number of respondents |

### Example 10

|  |  |
| --- | --- |
| Data | Syntax |
| .   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **K1** | **K2** | **K3** | **GENDER** | | 1 | 15 | 10 | . | male | | 2 | 0 | 5 | . | male | | 3 | 5 | . | . | female | | 4 | 10 | . | 5 | female | | 5 | . | . | . | female |   . | !stable type = average/  vars = k1 k2 k3/  col1 = gender/  options = share/  label = Share of volume. |
| Resulting output | Comments |
|  | * Because of that not all of the respondents did answered, number that refers to size of sample is missing: basis and „N“ row is empty * As the sum of variable K1 equals 30, and sum of K2 equals 15 and sum of K3 equals 5, total sum is 50. Share of variable K1 is than 60% (30/50), etc. |

### Example 11

|  |  |
| --- | --- |
| Data | Syntax |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | **A1** | **GENDER** | **AGE** | **SAMPLE1** | **SAMPLE2** | | 1 | apple | male | young | 1 | 0 | | 2 | apple | male | young | 1 | 0 | | 3 | pear | female | young | 1 | 0 | | 4 | pear | female | old | 0 | 1 | | 5 | peach | female | old | 0 | 1 |   . | !stable type = simple/  vars = a1/  col1 = gender/  col2 = age/  filter = sample1 sample2/  options = count. |
| Resulting output | Comments |
|  | * In tables frequencies are shown, and not column percentages, because in options there is count. Row with 100% is missing now (in this case it is same as N row). * There are two different tables in which are two different samples. Argument filter defines two binary variables, sample1 and sample2. In first table just respondents with no zero values on variable sample1 are counted, and on second table just respondents with no zero values on variable sample2 are used. * Basis in this case is 40% is calculated regarding the second table in which there are 2 of 5 respondents * As in second sample there is no “young” respondents so the column “Young” is not shown |

# Defining aliases

Alias is used for easy reference of variable lists in syntax. It is convenient to define list of variables in one place (and edit it if needed) and then to use that list in following commands.

## Syntax

!set name = <*alias names*>/

vars = <*names of variables in set*>.

## name

Alias name starts with double dollar characters. If ‘$$’ are omitted they will be automatically appended.

## vars

This argument represents list of variables that alias will consists of.

Example:

!set name = trik/

vars = d1 d2 d3 per $good b2 b3.

This syntax produces variable $$trik that can be used in standard !stable syntax, for example:

!stable type = simple/

vars = a1/

col1 = $$trik.

# !Set command

Necessity of repeating commands for any table definition is possible to define in syntax.

!set decimal= .

!set color= .

!set srtvl= .

!set ltperc= .

!set value= .

!set options= .

!set missing= .

There is !unset too, to delete this global !set.

Options and missing are different, because !unset in options will delete only specified options.

!unset options. Will delete all.

Local options like in

*!stable type = Simple /*

*vars = R4/*

*options = count/*

*col1 = stratum.*

are added to global, to turn off please specify $ before options.

!set options=count.

*!stable type = Simple /*

*vars = R4/*

*options = $count/*

*col1 = stratum.*

Will turn off count for that table.

Missing is similar, there is !set missing= . global definition, and then local in !stable definition which is added on global definition. There is no $ here because there is NORF, NODK, NONE…

# Selecting cases

We can select cases for all tables:

!select cases=*logical expresion/binary variable.*

!select cases= dem1==1. (Select all respondents with 1 on dem1, notice the == for equal

~= is unequal.)

As !select is macro, then / whill mean the end of argument, so if you want to say

!select cases = income/hsize >10.

Put . in front of /, so correct is

!select cases = income./hsize >10.

**!select of = (variable name) (condition) .**  turns off filter.

!select off = dem1==1.

# !split command

!split by lets repeat same set of tables for any value of chosen question.

It's very useful command in case of testing many samples. For example one respondent has tested one of three products and has answered to few questions connected. By using !split by we could describe only one block of tables and repeat it three times.

!split by=*variable*/code=**''***code to run***''**.

So for each different value of variable code will be run

Example:

!split by=age/code=''

!stable type=simple/

vars=a2\_1/

col1=dem1>tip/

options=descending. ''.

notice the . ''. on the end, first full stop is for code, then '' for end of code and then full stop for end of !split.

# Boxes

Now there is a possibility to define personal boxes.

For example if our scale has got 10 answers we receive Bottom 3 boxes and Top 4 boxes.

We receive frequencies and boxes (*!stable type=simple*).

!stable type = Simple/

vars = a1/

col1 = gender/

boxes = "Bottom 3 boxes" 1:3 "Top 4 boxes" 7:10 /

color = 2.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Total** | **Gender** | |
|  |  | Male | Female |
| N | 64 | 25 | 39 |
| sig |  | 0.99 | |
| Mark 1 | 09.4 | 08 | 10 |
| Mark 2 | 14.1 | 04 | 21 |
| Mark 3 | 17.2 | 20 | 15 |
| Mark 4 | 17.2 | 24 | 13 |
| Mark 5 | 03.1 |  | 05 |
| Mark 6 | 09.4 | 08 | 10 |
| Mark 7 | 10.9 | 08 | 13 |
| Mark 8 | 12.5 | 24 | 05 |
| Mark 9 | 03.1 | 04 | 03 |
| Mark 10 | 03.1 |  | 05 |
| Bottom 3 boxes | 40.6 | 32 | 46 |
| Top 4 boxes | 29.7 | 36 | 26 |
| Total | 100% | | |

/options=mean

/valid\_values=1 2 3 4 5 6 7 8 9 10.

shows row with average value.

# Weighting in script

There is a possibility of turn on and turn off weighting in script:

Turn on weighting:

!weight by=rim\_w\_3.

Turn off weighting:

!weight off.

By default !weight script will produce output, current weight status, to turn it off use print option:

!weight off/ output=no .

# How to use TABOFFREQ command

!taboffreq

vars = <variable names>/

filter = [<name of filter variable> | all]/

missing = [RF | DK | NORF | NODK | <*one number*> | <*more than one number*>] | NONE /

options = <ca> <nobonf> <sigc> <total> <whole> <sum> <count> <mean> <sd> <boxes> <nps>/

brkat= <number of categories>, *default value for this argument is 5*/

layout= <columns> <rows>/

color = [0 | 1 | 2 | 3]/

label=<title>.

Taboffreq produces table with frequencies for each category of variables defined in vars; frequencies are presented in rows (default) or columns. All variables defined in vars must have the same answer categories.

### options =

### ca

ca produces coordinates and labels for correspondence analysis which should only be copied in proper graph; this option can be used only in combination with taboffreq command.

See example 12

### sum

sum two more categories will be added, sum of all “negative” and sum of all “positive” answers.

### mean

adding mean of varijables on the bottom of table

### sd

adding standard deviation of varijables on the bottom of table

### sigc

Comparing columns among each other – multiple comparison test.

### nobonf

Disables Bonferroni correction for multiple-comparisons.

### boxes

Custom row sums, explained later.

### nps

Adds row with NPS based on 0-10 marks.

### Example 12

|  |  |
| --- | --- |
| Data | Syntax |
| .   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **N** | **Brand 1** | **Brand 2** | **Brand 3** | | (Image) statement 1 | 22 | 18 | 8 | 3 | | (Image) statement 2 | 24 | 23 | 8 | 8 | | (Image) statement 3 | 7 | 3 | 5 | 2 | | (Image) statement 4 | 11 | 1 | 7 | 3 | | (Image) statement 5 | 19 | 10 | 14 | 0 | | (Image) statement 6 | 21 | 0 | 18 | 6 | | !taboffreq  vars = $s1 $s2 $s3 $s4 $s5 $s6/  options = ca.  \* Before running this syntax, syntax for multiple response set should be executed (s1 – s6) |
| Resulting output | Comments |
| |  |  |  |  | | --- | --- | --- | --- | | -0.845 | Brand 1 | -0.099 |  | | 0.659 | Brand 2 | -0.306 |  | | 0.315 | Brand 3 | 1.082 |  | | -0.629 | (Image) statement 1 |  | -0.145 | | -0.606 | (Image) statement 2 |  | 0.433 | | 0.282 | (Image) statement 3 |  | 0.145 | | 0.870 | (Image) statement 4 |  | 0.392 | | 0.066 | (Image) statement 5 |  | -0.943 | | 1.163 | (Image) statement 6 |  | 0.176 | | * Table presents coordinates and labels that should only be copied in proper graph * Data that were used for producing coordinates for correspondence analysis represent frequency of matching (image) statements with target brands |
| Data | Syntax |
| .   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | **N** | **Mark 1** | **Mark 2** | **Mark 3** | **Mark 4** | **Mark 5** | **Sum -** | **Sum +** | | Attribute 1 | 800 | 16.3 | 16.5 | 26.0 | 17.0 | 24.2 | 32.8 | 41.2 | | Attribute 2 | 800 | 7.6 | 9.0 | 21.2 | 26.0 | 36.2 | 16.7 | 62.2 | | Attribute 3 | 800 | 27.8 | 15.9 | 26.1 | 15.6 | 14.7 | 43.6 | 30.3 | | Attribute 4 | 800 | 9.5 | 16.7 | 28.5 | 23.1 | 22.2 | 26.2 | 45.4 | | Attribute 5 | 800 | 8.0 | 12.4 | 27.7 | 25.9 | 25.9 | 20.4 | 51.8 | | Attribute 6 | 800 | 33.6 | 13.1 | 20.9 | 16.9 | 15.5 | 46.8 | 32.3 | | !taboffreq  vars = a1 a2 a3 a4 a5 a6/  options = sum/  brkat=5. |

# How to define matching labels with specific colors

In order to make observing results easier and to present the same category (i.e. the same label) with the same color in each graph, it is necessary to do the following steps:

* Pick the list of labels that should be always presented with the same color (note: picked labels must be the same as in data base)
* Specify color for each picked label
* Define global variables IsteBojeLabel and IsteBojeNum with these data (i.e. picked labels and specified colors)

Firstly, it is necessary to run command IsteBoje (type and run this command in syntax window).

Resulting output of this command is list of all labels (i.e. strings) in target data base; labels (i.e. strings) are presented under quotes. This list must be used in the same form (i.e. the same case and the same spaces) as it is presented in the resulting output. Only in that way, program will recognize specified labels.

Resulting list of labels presents all labels in the target data base. This list can be shorten if we do not want to specify colors for all these labels. Those labels that are not picked will, by default, get randomly picked colors.

In the file BojeGrafici v2.xls there is list of colors, with ordinal number, i.e. ID number for each color. Each (picked) label should be matched with the ID of color that we want to have in graphs which present this label.

Final step is defining global variables IsteBojeLabel and IsteBojeNum and assigning them values; this should be done in syntax window.

Example:

global IsteBojeLabel, IsteBojeNum

IsteBojeLabel=[ "063 (062)"

"064 (065)"

"MTS"

"MTS - post paid"

"MTS - pre paid"

"Mobilkom"

"Mobtel"

"Mobtel - post paid"

"Mobtel - pre paid"

"New third (Mobilkom - pre paid)"

"Telekom"

"Telenor"

"Telenor - post paid"

"Telenor - pre paid"

"Telenor/Mobtel"

"mts"];

IsteBojeNum = [11

3

3

3

3

44

5

5

5

44

3

11

11

11

11

3];

Values in variable IsteBojeLabel (put in square brackets) are strings that were result of running command IsteBoje, and values in IsteBojeNum (put in square brackets) are ID numbers of colors in BojeGrafici v2.xls file.

This code should be run only once, at the beginning of work in the program (i.e. when we start program).

# How to define multiple variables

## Syntax

!multiple name = <*variable names*>/

vars = <*names of variables in multiple set*>/

label = <*label for new variable*>/

[value = <*list of values*>].

## name

Name of multiple variable should follow all standards in naming variables (i.e. the same standards as in case of naming simple variables). Taking into account that multiple variable name starts with sign „$“, this name should not be longer than 7 characters. If sign „$“ is not specified, it will be automatically assigned to variable name.

## vars

This argument represents list of variables that multiple variable will consists of.

## label

Represents label for new multiple variable. This label should not consist of special characters, such as: dot, equal, slash, percent (. = / %), or should be double quoted.

## Value

If the list of values is specified, then multiple variable is binary and specified values are considered as “Yes” answers, i.e. answers to be counted. It is possible to specify more than one value (in contrast to SPSS where only one value can be specified).

NOTE: Binary variable defined with more than one value, will not be computed in that form in SPSS, only the first specified value will be considered.

Example:

!multiple name=posed/

vars=dem9\_1 dem9\_2 dem9\_3 dem9\_4 dem9\_5 dem9\_6 dem9\_7 dem9\_8 dem9\_9 dem9\_10/

label=Durable goods/

value=1.

This syntax produces variable $POSED that can be used in standard !stable syntax, for example:

!stable type = simple/

vars = $posed/

col1 = tip.

This syntax is the same as the following:

!stable type = binary/

vars = dem9\_1 dem9\_2 dem9\_3 dem9\_4 dem9\_5 dem9\_6 dem9\_7 dem9\_8 dem9\_9 dem9\_10/

col1 = tip/

value = 1.

# How to define block multiple variables

Block multiple variables are multiple variables that have all characteristics of standard multiple variable but are also attached to another multiple variable, i.e. paired variable by variable. Usually, this is data record that should be split up in a new data base and it is mostly done by making new data base where records represent all given answers in multiple variables (so called “underlined” data base).

Take the following data set as an example:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **A1** | **B1** | **C1** | **A2** | **B2** | **C2** | **A3** | **B3** | **C3** | **GENDER** |
| 1 | Apple | Glass | 2 | Apricot | Plastic | 2 | Plum | Can | 2 | Male |
| 2 | Apple | Plastic | 3 | . | . | . | . | . | . | Male |
| 3 | Pear | Can | 1 | . | . | . | . | . | . | Female |
| 4 | Pear | Glass | 10 | . | . | . | . | . | . | Female |
| 5 | Peach | Glass | 1 | . | . | . | . | . | . | Female |

In this table, there are five respondents; A1, A2 and A3 are variables included in multiple variable $A and they represent answers on a question “Which fruit juice have you drunk during the past 7 days?”. Variables B1, B2, B3 consist multiple variable $B which represents answers on a question “In which type of package was the juice that you have drunk?”. It is clear that values in B1 are attached to values in A1, values in B2 to values in A2, and so on. Variables C1, C2 and C3 consist multiple variable $C, that is also attached to previous two multiple variables ($A and $B) as it represents answers on a question “How many glasses of that juice have you drunk?”.

If we want to see connection between type of fruit juice and type of package, it is clear that we can not only cross over multiple variables $A and $B.

|  |  |
| --- | --- |
| !multiple name=$A/  vars=a1 a2 a3/  label=Fruit.  !multiple name=$B/  vars=b1 b2 b3/  label=Package.  !stable type = simple/  vars = $a/  col1 = $b. |  |

In the resulting table, we can see that each type of fruit (i.e. juice) occurred in glass type of package, which is not correct. First respondent used glass package, but only in case of apple juice. It is important to notice that in this case sample size was 5 respondents and what we actually wanted was to cross over type of juices with type of package. In order to do so we have to define these variables as blockmultiple.

|  |  |
| --- | --- |
| !multiple name=$A/  vars=a1 a2 a3/  label=Fruit.  !multiple name=$B/  vars=b1 b2 b3/  label=Package.  !blockmultiple vars = $a $b.  !stable type = simple/  vars = $a/  col1 = $b. |  |

In this way we crossed over multiple variables correctly: only apple, pear and peach were consumed in glass package and we also have number of records which is the same as total number of consumed fruit juices, 7.

Of course, it is possible to use quantitative multiple variables as well. These variables have full meaning when they are used in combination with another block multiple variables. In case we want to get multiple variable average, target variable will be replaced with sum of all variables that consist multiple variable, i.e. multiple variable average is average of sum of all variables in multiple set.

|  |  |
| --- | --- |
| !multiple name=$A/  vars=a1 a2 a3/  label=Fruit.  !multiple name=$B/  vars=b1 b2 b3/  label=Package.  !multiple name=$C/  vars=c1 c2 c3/  label=Quantity.  !blockmultiple vars = $a $b $c.  !stable type = average/  vars = $c/  col1 = $b/  decimal = 1. |  |

We got 7 records again (all consumed juices) and average of variable “Quantity”, in total and per type of package.

In case we want to get average of block multiple variable, which is in fact quantitative variable, per categories of simple variable or multiple variable which is not in the same block with this variable, we get average of sum of that multiple variable. For example:

|  |  |
| --- | --- |
| !multiple name=$A/  vars=a1 a2 a3/  label=Fruit.  !multiple name=$B/  vars=b1 b2 b3/  label=Package.  !multiple name=$C/  vars=c1 c2 c3/  label= Quantity.  !blockmultiple vars = $a $b $c.  !stable type = average/  vars = $c/  col1 = gender/  decimal = 1. |  |

Average is 4.2 and it is computed as average of the following values: 3, 1, 10, 1 and 6 (for the first respondent, where we have sum of answers given on 3 variables)

## Syntax

!blockmultiple vars = <*multiple variables names*>.

All multiple variables in one block have to be consisting of the same number of variables. It is not possible to cross over multiple variables from one block with multiple variables from other variables.

# How to use BIGTAB command

bigtab is basic command for producing tables. stable command actually uses bigtab command for producing tables.

## Syntax:

!bigtab row = <*name of variables that will be presented in rows of table*> /

col = <*name of variables that will be presented in columns of table*> /

statistics = count

popXYZ

rowpct

colpct

totpct

poppct

mean(<*variable name*>)

trimmean(<*variable name*>)

median(<*variable name*>)

min(<*variable name*>)

max(<*variable name*>)

range(<*variable name*>)

sd(<*variable name*>)

var(<*variable name*>)

ciXX(<*variable name*>)

trimciXX(<*variable name*>)

index

colshare[(<*variable name*>)]

trimcolshare[(<*variable name*>)]

rowshare[(<*variable name*>)]

trimrowshare[(<*variable name*>)]

layout = [statisticsinrow | statisticsincol] /

ltperc = <*two numbers*> /

options = noprint

samplesize

ownbase

nozerocolumns

indexcolor   
 nps/

marks = <*number of marks*> /

index = < *variable name*> <value> /

filter = [<*name of filter variable*> | all].

## row, COL

Name of variables that will be presented in table rows i.e. columns. It is possible to use multiple variables or system variable $total that marks all respondents (i.e. records).

Variables will be presented in the named order, one by one. If we want to nest two variables specified in columns, than we use option “>”.

## statistics

This argument stands for list of statistics that should be presented in table. There are two types of statistics, “frequency” and “summarizing” statistics.

For “summarizing” statistics, brackets should include variable that statistic should be applied to (for example, mean(dem2) gives average; mean is summarizing statistic for dem2 variable)

### count

Shows frequencies of variables in rows, i.e. columns.

### popXYZ

Shows frequencies in population parameters. Population size is defined by option pop (i.e. by typing population size right after pop), for example pop2340 means that population size is 2340 thousands (due to error, it is desirable to define population size in thousands).

It is possible to use global variable popsize; we have only to define global variable and assign it a value at the very beginning.

Example

global popsize;

popsize = 2340;

### rowpct

Row percentage.

### colpct

Column percentage.

### totpct

Percentage referring to total, i.e. the whole sample.

### poppct

Percentage referring to population; very similar to previous value. This option has its full purpose when we do cross tabulation with time variable, i.e. variable which name starts with @ character. Imagine that we have variable referring to the month when the research was conducted (i.e. typical time variable) and that we cross over this variable with variable “gender”. Of course, male – female ratio (app. 1:1) should be the same in each month, i.e. there should be 50% of male in each month, not 50% male/(number of months).

### mean

Average.

### trimmean

Trimmed average, computed on central 90% of values. 5% of upper and 5% of lower values are not included in average.

### median

Median of target variable.

### min

Minimum of target variable.

### max

Maximum of target variable.

### range

Range of target variable (i.e. maximum minus minimum).

### sd

Standard deviation of target variable.

### var

Variance of target variable.

### ciXX

Confidence interval of target variable where XX stands for significance level, for example ci80(dem2) shows lower and upper bound of 80% confidence interval for dem2 variable.

### trimciXX

The same as previous option, trimciXX represents confidence interval, but in this case CI is calculated on trimmed statistics (excluding upper and lower 5% of values). XX stands for significance level.

### index

Shows index values referring to variable category which is specified in index argument (if variable category is not specified, index will be calculated referring to the first column in table). Index represents ratio of two column percents multiplied with 100.

### colshare , colshare (variable name)

Column share. This can be “frequency” as well as “summarizing” statistic. If variable is not defined, then frequency shares are calculated (useful in case of multiple variables). If variable is not defined then shares from answer frequencies are calculated (makes sense if multiple answer variable is tabulated), and when variable is defined then shares based on that variable are calculated. In case the layout of statistics is in rows, other variables for which we want to have column shares are becoming part of that share as well.

### trimcolshare(variable name)

The same as colshare, except this has to be “summarizing” statistic (it is necessary to define variable). Shares are calculated on trimmed values.

### rowshare, rowshare(variable name)

Row shares. This can be “frequency” as well as “summarizing” statistic. If variable is not defined, then frequency shares are calculated (useful in case of multiple variables). If variable is not defined then shares from answer frequencies are calculated (makes sense if multiple answer variable is tabulated), and when variable is defined then shares based values of that variable are calculated. In case the statistics layout is in columns, other variables for which we want to have column shares are becoming part of that share as well.

### trimrowshare

The same as rowshare, except this has to be “summarizing” statistic (it is necessary to define variable). Shares are calculated on trimmed values.

## layout

Usually, statistics are presented in rows, and it is possible to present them in columns by using statisticsincol option.

## ltperc

This argument consists of two numbers, first number represents percent which will be used as benchmark for recoding all values (in column “total”) lower than this benchmark in value of another number in this argument.

If value of another number in this argument has not its label in SPSS data base, it will be shown as number only.

## Options

### noprint

Option defining not to print table.

### samplesize

If we do expansion (i.e. population projection) by data weighting, then sum of weighting coefficients will be higher than factual sample size. In order to calculate significance basing on sample size, which is standard, option samplesize should be used.

### ownbase

Usually, binary tables are computed by taking the same base for all target variables. If there are five answers on the first variable and 3 answers on the second variable, than higher base will be used for calculations on the second variable as well. This means that even if all respondents give positive answer on second variable, in total it can be maximally 60% (3/5\*100). If we want each variable to have its own base, than we have to use ownbase option. In that case, second variable in our example will have 100% of positive answers, because all three respondents gave positive answers. If we use this option when variables have different bases, line “N”, i.e. line “unwN” will be empty.

### nozerocolumns

All target table columns where subsample size (line N or unw N) equals 0 will not be presented in table. This option is useful when nesting tables because it produces more compact tables wilth smaller number of columns.

### indexcolor

For index statistic, colors cells by following (arbitary) principle:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| >=150 | >=130 | >=115 | jhgjhg | <=85 | <=70 | <=50 |

The idea is to emphasize differences in case N is too small for obtaining statisticly significant differences.

### nps

Returns the NPS score for the variable specified.

## marks

Adds up summarizing rows for the first and for the second half of rows number defined in marks option, if defined number of rows is even number. If defined number of rows is odd number than middle value will be excluded from summarizing calculation.

In case we defined marks option with value 5, than following rows will be added up in table: row summarizing answers 1 and 2 and row summarizing answers 4 and 5. More over, there will be row representing average values for scale defined in marks; in case of 5 point scale, average will be calculated only for variable values 1 to 5, all other values will be presented as percentages, but will not be included in average calculation.

## index

In this argument, we define name of variable and value of variable category that will be used for index calculation in table. For example index = dem1 2/ will produce table with indexes referring to values that we got for female subpopulation (dem1 is variable gender, and 2 stands for female).

## Select

It is possible to define numeric expression, and if that expression evaluates at 0 the record will be excluded from analysis. See this argument in !stable macro explanation.

### Example 13

|  |  |
| --- | --- |
| Data | Syntax |
| |  |  |  |  | | --- | --- | --- | --- | |  | **A1** | **B1** | **C1** | | 1 | apple | male | young | | 2 | apple | female | young | | 3 | apple | male | young | | 4 | apple | female | young | | 5 | pear | female | young | | 6 | pear | male | old | | 7 | pear | male | old | | 8 | pear | female | old | | 9 | peach | female | old | | 10 | peach | male | old |   . | !bigtab row = a1/  col = gender age/  statistics = colpct/  index = age 1/  layout = statisticsincol. |
| Resulting output | Comments |
|  |  |

### Example 14

|  |  |
| --- | --- |
| Data | Syntax |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | **A1** | **B1** | **C1** | **D1** | **GENDER** | | 1 | apple | glass | 2 | 3 | male | | 2 | apple | glass | 7 | 3 | female | | 3 | apple | plastic | 3 | 4 | male | | 4 | apple | can | 2 | 2 | female | | 5 | pear | glass | 10 | 44 | female | | 6 | pear | glass | 5 | 33 | male | | 7 | pear | plastic | 5 | 2 | male | | 8 | pear | can | 1 | 21 | female | | 9 | peach | glass | 1 | 32 | female | | 10 | peach | plastic | 3 | 3 | male | | 11 | peach | can | 8 | 11 | female | | 12 | apple | glass | 2 | 3 | male |   . | !bigtab row = a1 b1/  col = gender $total/  statistics = colpct mean(c1) sd(d1). |
| Resulting output | Comments |
|  |  |