***Series Response tools***

SPSS macros by Kirill Orlov

kior@akado.ru, ttnphns@gmail.com

<https://www.spsstools.net/en/KO-spssmacros>

All rights reserved

*Tools for series of items.* Collection of macros for a “simple matrix question”, i.e., a series of variables with a common pool of alternative responses (Single response series, SRS); for example, a set of items each scored by rating scale or ranked. One of the macros is for the data respondents ranked and it shifts the variables into the categorical multiple response set or back. Another macro is intended for more general tasks of translating values and variables into each other as well as for calculating on reduplicating values. The third macro is for a situation when respondents rated not all items but only those they had chosen before, and the rating data having been entered in a packed (quickened) mode.

* If there had been a question asking to rank or to select-and-rank, then whichever way it was entered – in the form of series of items or in the form of categorical multiple response set, [!KO\_RANKREV](#_МАКРОС_!RANKREV:_ПЕРЕВОД) will recode one entry into the other.
* If you face a more general aim of the type “make variables out of values” or “make values out of variables”, use [!KO\_VALVARS](#_МАКРОС_!VALVARS:_ПЕРЕВОД). This macro is useful also to reckon some stats horizontally, especially under codition banning to account some data cells.
* Manual data entry of a question like “select, and rate selected items” poses a problem (one has to search for the needed variable where to enter the rating value). Macro [!KO\_SRSREF](#_МАКРОС_!SRSREF:_РАЗВЕРТКА) solves it. You may enter ratings consecutively (“packed”) – the macro will unwrap this entry into the correct set of variables.

**Single response series** (SRS) is a set of items or subquestions with a common bank of response variants; on each item, only one response can be given. The other name of such a set is simple matrix question. A typical example is items, each rated by a scale (like Likert, Osgood, or else) or simply “Yes-No”. Another example may be questions on ranking, where items need to be distributed among ordinal places. Variables in SRS series are items, and values are response codes, rating scale scores, or rankings.

# MACRO !KO\_RANKREV: CONVERTING VARIABLES WITH RANKS INTO VARIABLES-RANKS OR BACK

Version 2, Apr 2008 (Version 1, Jan 2001). Tested on SPSS Statistics 11, 13, 26.

!KO\_rankrev vars= *v1 v2 v3 v4* /\*Name-by-name set of variables to recode

/capnum= *w\_ 3* /\*Prefix into names of new variables;

/\*after it may specify their number (equal to the higher value in VARS);

/\*If unspecify CAPNUM, then VARS themselves will be recoded

/filler= /\*Optional: value into not filled cells of new variables

/\*(it will be user-missing).

Minimal specification VARS.

Data of a ranking question (to distribute some items or variants by ranks: 1st, 2nd and so on) may be entered in twofold manner. In one way – as a series of items (SRS set), i.e., variables correspond to items and data in them are the occupied places, ranks; in the other way – as a categorical multiple response set (MRC), where variables are response “attempts”, ordered (and in our case it is the rank assigned), while data in the variables are codes enumerating the items. For example, if there were items (some stimuli) numbered as **1, 2, 3**, and a respondent assigned them ranks or places, correspondingly, *3, 1, 2,* then the enter packing in the form of SRS will be:

|  |  |  |
| --- | --- | --- |
| **var1** | **var2** | **var3** |
| *3* | *1* | *2* |

and in the form of MRC it will be:

|  |  |  |
| --- | --- | --- |
| *var1* | *var2* | *var3* |
| **2** | **3** | **1** |

In the first instance the ranks (the data) are the values, and in the second instance they are the variables.

The MRC-way of data entry is more parsimonious in a situation where the question asks to rank not all items but only those which the respondent will select it is the so-called multiple response question with ranking), – because then MRC can consist of fewer number of variables than there exist items. From the data analysis point of view, MRC set can be processed to tabulation as a multiple response set, i.e., without taking ranks into account, or it can be analyzed for distribution of items within the 1st place, the 2nd place, and so on, separately. SRS set, on the other hand, allows to analyze variables as ordinal or scale data, for example to observe the mean rank obtained for each item.

This macro transforms one structure of a “ranking-of-items question” into the other structure, that is, SRS set into MRC set or vice-versa. You may request to create new variables or to alter the input variables themselves. Values in the input variables must be ordinal numbers 1, 2, 3…, and a value should encounter in a case no more than once[[1]](#footnote-1). (If that is not so, use more universal macro !KO\_VALVARS.) Macro !KO\_RANKREV (“ranking reverse”) does so: if in the i-th input variable, as they are listed in VARS, the case’s value = j, then, in the j-th output variable, it will be = i.

***Subcommands***

**VARS**

Specify name-by-name a set of variables which are to recode into “the opposite” structure. They can be an MRC set or its part, or an SRS set or its part. Variable names can be any. Values of the input variables must be natural numbers (positive integers) 1, 2,…. Some values of this series may really be absent in the data. All other values except natural numbers, as well as missing values, are ignored by the macro. Values exceeding the upper limit specified in CAPNUM s/c are ignored; by default this limit equals the number of variables VARS.

Values coming out in the output variables are the sequential numbers of the input variables VARS. Output variables are assigned scale level of measurement; however actually they are either ordinal (if they are SRS) or nominal (if they are MRC).

**CAPNUM**

Specify a prefix into the names of output variables, you may take it in quotes/apostrophes (they are necessary if the separator-period is used as the end of the prefix: ‘var.’). Output variable names will be formed of the prefix and the ordinal numbers 1, 2,…, - which are values in the input variables. After the prefix, you may indicate the “upper limit” – an integer equal to the maximal value existing in or interesting you in the input variables, - and that many output variables will be created. By unspecifying such number after the prefix, the macro will set it equal to the number of input variables.

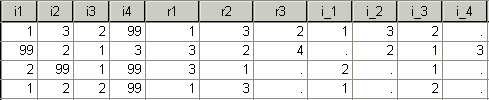
Output variables are created new. No such named variables should be in your dataset. (If some of the being created variables already exist in the dataset, SPSS will issue an error. Please check the variables then – whether they were or were not modified by the macro.)

You may omit specifying CAPNUM or omit the subcommand. In this case, input variables themselves will be the output ones: the macro will recode them, completely renewing. Value labels and user-missing specifications will be erased. Variable labels will remain as they are. When CAPNUM is omitted, the “upper limit” mentioned earlier is implied the default one.

EXAMPLE 1. In a ranking question, 4 objects (items) were presented. Respondents had to select up to 3 of them and to rank those by importance (1st place, 2nd, 3rd). The data were initially entered as SRS set, i.e., variables *I1 – I4* are the items, and values in them are the ranks.

!KO\_rankrev vars= i1 i2 i3 i4 /capnum= r 3.

!KO\_rankrev vars= r1 r2 r3 /capnum= i\_ 4.



* The first command translates SRS to MRC consisting of variables *R1, R2, R3*, which are three: this number was requested, as there was allowed to rank up to three items. *R1* is “rank 1”, *R2* is “rank 2”, *R3* is “rank 3”. Values in them – the items from 1 to 4, in correspondence with the input variables. In case 4, rank 2 encountered twice, - so only one, last its encounter was considered.
* The second command does the back translation of the MRC to SRS, in new variables *I\_1, I\_2, I\_3, I\_4*. Number 4 indicates that 4 variables should be created, as there were 4 items (with codes 1 to 4 in the MRC).

**FILLER**

By default, the macro leaves empty cells in the output variables empty (system-missing). Here you may specify a filler code (any number) to insert there. It will be given user-missing status. The macro does *not* insert the filler into cases that are missing (user- or system-missing) in *all* the VARS variables; such cases are output system-missing.

***Special regimes***

The macro does not respond to weightedness, filteredness (FILTER, USE), splitness of the dataset. It doesn’t obey temporary (under TEMPORARY) transformations.

# MACRO !KO\_VALVARS: VALUES-VARIABLES CONVERSION INTO EACH OTHER

Version 2, Sep 2004 (Version 1, Mar 2001). Tested on SPSS Statistics 11, 13, 26.

!KO\_valvars vars= *v1 v2 v3 v4* /\*Input numeric variables, name-by-name

/values= *1 to 8* EXCEPT *3 5* /\*List of values (nonnegative numbers, either name-by-name,

/\*or by range via “to”) you are interested in VARS; after range may

/\*enumerate, after k\w EXCEPT, values to exclude from the range

/newvars= 'w\_' /\*Names for new variables: name-by-name (as many as VALUES) or single prefix

/newvals= EXTRACT /\*Values for new variables: 1) enumerate (as many as VARS): nonnegative

/\*numbers and/or variable names;

/\*2)ORDINAL – generate integers 1,2,...,number of VARS;

/\*3)EXTRACT – take from names of VARS; 4) ALLONE – all values 1

/filler= 99 /\*Optional: value into not filled cells in the new variables

/\*(will be user-missing)

/miss= SET /\*If in created variables you need missing cases: make it sysmis, if

/\*1) it is missing in all VARS (SET);

/\*2) it is missing in this variable (not necessarily of VARS):

/\*(VAR variable name)

/redupl= MINGAP /\*If a value (of VALUES) may encounter in VARS more than once,

/\*then which to account:

/\*FIRST, LAST (default), RANDOM, MAX, MIN, reckon all:

/\*MEAN, SUM, RANGE, MINGAP, MAXGAP (see).

Minimal specification VARS, VALUES, NEWVARS, NEWVALS.

The macro is intended for tasks that, in general terms, may be understood as “turning values into variables, and variables into values”. Say, when variables are items and values are responses, you may convert to such a built where variables are responses and values are items. Or convert backward. This is a more general macro than !KO\_RANKREV, and designed not just for ranking data and offering more options. It is applicable also to compute some statistics horizontally, that is, within cases. The macro creates always new variables.

The gist of the macro’s job is as follows.

Let there be *p* input variables VARS: *VAR1, VAR2,…, VARp,*

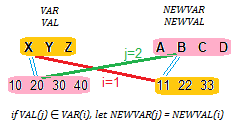
in which you are interested in the list of *k* values VALUES: *val1, val2,…, valk.*

You specify, in correspondence with the input variables, *p* new values[[2]](#footnote-2) NEWVALS *newval1, newval2*,…, *newvalp,*

and in correspondence with the input values – *k* new (not existing so far) variables NEWVARS: *NEWVAR1, NEWVAR2,…, NEWVARk.*

The macro creates those *k* variables – such that if, within a respondent (case), value *valj* belongs to variable *VARi*, then variable *NEWVARj* receives value *newvali* (**Fig. 1**). Values VALUES enter the names of variables NEWVARS.

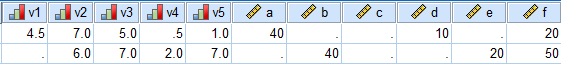
If the respondent has multiple instances of the value in the input variables (i.e., *valj* belongs to several variables VARS), the macro possesses a number of options to account/reckon them (see s/c REDUPL).

****

**Fig. 1.** Variables are dataset columns and values are case’s (row’s) data. The macro does so that to the input variables, output values correspond, and to input values, output variables correspond.

EXAMPLE 1.

!KO\_valvars vars= v1 v2 v3 v4 v5 /values= .5 2 3 4.5 6 7 /newvars= a b c d e f /newvals= 10 20 30 40 50.



* In variables *V1-V5* (five), values .5, 2, 3, 4.5, 6, 7 (six) interest us. New variables *A-F* (six) - let have possible values 10, 20, 30, 40 ,50 (five).
* Consider the 1st case. Value .5 (the 1st in the list) is found in variable *V4* (which is the 4th in the list), therefore in the 1st new variable (*A*) there will be the 4th new value (40).
* No values 2, 3 are found in the 1st case. Value 4.5 (the 4th in the list) is found in variable *V1* (the 1st in the list), therefore in the 4th new variable (*D*) there will be the 1st new value (10).
* No value 6 is found in the 1st case. Value 7 (the 6th in the list) is found in variable *V2* (the 2nd in the list), therefore in the 6th new variable (*F*) there will be the 2nd new value (20).
* In the 2nd case, value 7 encounters more than once, i.e., is found in different variables. By default of s/c REDUPL, the macro accounts for the last encounter of the value. Therefore new variable *F* gets new value 50.

EXAMPLE 2. This example reproduces the job of macro [!KO\_RANKREV](#_МАКРОС_!RANKREV:_ПЕРЕВОД) shown there in EXAMPLE 1.

!KO\_valvars vars= i1 i2 i3 i4 /values= 1 to 3 /newvars= r /newvals= ORDINAL.

!KO\_valvars vars= r1 r2 r3 /values= 1 to 4 /newvars= i\_ /newvals= ORDINAL.

* The first command converts variable set *I1, I2, I3, I4*, in which we are interested in values 1, 2, 3, into the set of variables, respectively, *R1, R2, R3*. If suppose, *I2*=1 (one of VALUES), then at output there will be *R1*=2 (ordinal number of *I2* in the VARS list).
* The second command converts it back. It turns *R1, R2, R3,* in which we are interested in values 1, 2, 3, 4, into, respectively, *I\_1, I\_2, I\_3, I\_4*. If suppose, *R1*=2 (one of VALUES), then at output there will be *I2*=1 (ordinal number of *R1* in the VARS list).

***Subcommands***

**VARS**

Indicate name-by-name the set of input variables (two minimum). It can be any numeric variables.

**VALUES**

List of valid values of interest in VARS, one or more nonnegative numbers[[3]](#footnote-3). You can specify it either by enumeration or by a range via “to”: min to max, but not mixed. In the former case the values need not be integers. While when the values are specified via a range, the macro forms their list as integers from min to max, which themselves must be integers.

By insertion of keyword EXCEPT after the range you can specify a list of values that you want to omit from the range. For example, if VALUES= *1 to 5* EXCEPT *2 4*, then the list of values the macro will generate is *1 3 5*. If no codes follow the word EXCEPT, then the word is ignored.

**NEWVARS**

This is the list of output, new variables. They must not coincide in names with any variables existing in the dataset[[4]](#footnote-4). The number of variables in the list must equal the number of values VALUES, because these two lists are correspondent by elements. So specify the name-by-name list of new variables.

Instead of a name-by-name list, you may indicate a single prefix, optionally in quotes or apostrophes (they are necessary if the prefix ends with a period separator, “var.”). Then the names of the new variables will consist of that prefix and the VALUES themselves as endings.

Output variables receive scale measurement level. You may afterwards set them the level you want.

**NEWVALS**

This is the list values of the new variables. It and the list VARS are correspondent by elements. The list can be specified various ways:

*List of values and/or variables* - enumerate nonnegative numbers, as many as there are VARS variables, in correspondence to them. Values need not be all different. And you may indicate names of variables existing in the dataset, which contain the needed values (and not necessarily nonnegative). Specifying the list as variables means that for each different case, separate list of values is possible. Missing values in those variables, if any, will not be transferred to the new variables being created by the macro. Among the variables, variables of VARS may be mentioned.

ORDINAL - the macro forms the list automatically as natural numbers 1, 2, …, the number of input variables.

EXTRACT - the macro tries to extract values from the input variables’ names. That must be just integer numbers. The input variable names must end with these numbers-values. For example, variable named *VAR2* will yield value 2. Variable named *MRD3.2* will also give value 2. The macro picks an integer from the tail of a name.

ALLONE - this is the same as the value list of units. It makes sense along with REDUPL=SUM to count equal values in the input data.

**FILLER**

Optional subcommand. By default, the macro leaves empty (system-missing) cells left not filled in the output variables. Here you may indicate a filler code, any value, to insert there. It will be given user-missing value status.

**MISS**

Subcommand to account for / create missing values. By default/unspecifying and with MISS=NO, empty cells (system-missing) do not arise in the output variables, if you specified FILLER. MISS allows to make system-missing some cases. MISS makes a case system-missing in *all* the created output variables. Indicate one of the two conditions:

SET - a case will be system-missing if it has missing values (user-missing or system-missing) in *all* input VARS variables.

VAR *varname* - a case will be system-missing if it has missing value (user-missing or system-missing) in this specific variable (it can be any variable of the dataset).

MISS makes cases system-missing *after* the output variables with their data are created.

**REDUPL**

If a value of VALUES encounters in a respondent (dataset case) more than once in the input variables VARS when the latter are multiple, there arises the problem which of them to account: for corresponding values NEWVALS of them (of those VARS) may be different. In REDUPL subcommand, you specify how to account for duplicated values in VARS.

The following options account *one* of the tied (equal) values – a value belonging to a ith variable in VARS (consequently, the ith value of NEWVALS is used):

LAST - (default/unspecifying) account the last in the VARS line encounter of the value.

FIRST - account the first in the VARS line encounter of the value.

RANDOM - choose one of the equal values randomly[[5]](#footnote-5).

MIN - account the value of that VARS variable for which the NEWVALS value is least.

MAX - account the value of that VARS variable for which the NEWVALS value is greatest.

The following options account for *all* encounters of the value and take – as the NEWVALS value – a summary statistic of all their (the ties) corresponding NEWVALS values:

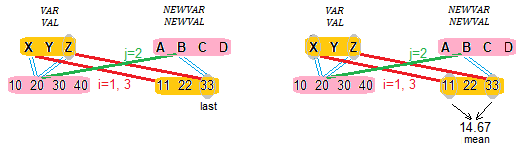
MEAN - mean of their NEWVALS values is computed.

SUM - mean of their NEWVALS values is computed.

RANGE - range of their NEWVALS values is computed. If a value is not tied (single encounter), this yields range 0.

MINGAP - minimal gap (difference) between their NEWVALS values is computed. If a value is not tied (single encounter), this yields gap 0.

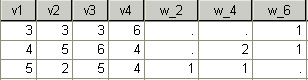
MAXGAP - maximal gap (difference) between their NEWVALS values is computed – after the latter are sorted ascendingly to consider differences among neighbours there. If a value is not tied (single encounter), this yields gap 0.



**Fig. 2.** REDUPL=LAST and REDUPL=MEAN. Value 20 encounters twice in VARS: in *X* and *Z*; therefore one has to select between NEWVALS 11 and 33, or to compute their summary statistic.

EXAMPLE 3. Counting of values horizontally. The macro plays the role of SPSS command Count.

!KO\_valvars vars= v1 v2 v3 v4 /values= 2 to 6 EXCEPT 3 5 /newvars= 'w\_' /newvals= ALLONE /redupl= SUM.



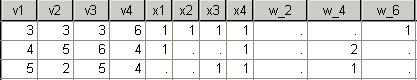
As seen from VALUES, the values of interest in the input variables are 2, 4, 6, to which output variables with prefix *W\_* correspond.

REDUPL orders to cumulate the sum of values NEWVALS at every encounter of a value of interest. Since NEWVALS=ALLONE (what = 1 1 1 1), it turns out to be counting of the values of interest instances.

EXAMPLE 4. The same example of counting of values, but with partial ban for counting.

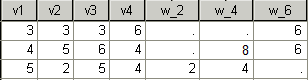
!KO\_valvars vars= v1 v2 v3 v4 /values= 2 to 6 EXCEPT 3 5 /newvars= 'w\_' /newvals= x1 x2 x3 x4

/redupl= SUM.



In NEWVALS subcommand, instead of ALLONE, variables *X1–X4* are indicated, which contain ones not everywhere: empty cells forbid to account their corresponding cells in variables *V1-V4*. Thus, not all occurrences of the values 2, 4, 6 in *V1–V4* have been counted.

EXAMPLE 5. Computation of horizontal sum of equal values.



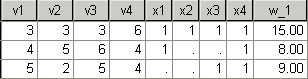
!KO\_valvars vars= v1 v2 v3 v4 /values= 2 to 6 EXCEPT 3 5 /newvars= 'w\_' /newvals= v1 v2 v3 v4

/redupl= SUM.

As NEWVALS, variables VARS themselves are specified. Because REDUPL=SUM, output variables *W\_2, W\_4, W\_6* will present summations of repetitions of values 2, 4, 6, respectively.

EXAMPLE 6. Computation of a horizontal statistic (for instance, sum of the values) with partial ban.

!KO\_valvars vars= x1 x2 x3 x4 /values= 1 /newvars= 'w\_' /newvals= v1 v2 v3 v4 /redupl= SUM.



In the focus of interest – variables *V1 – V4*, which this time are mentioned as NEWVALS. Units are stored in variables *X1 – X4*, specified as VARS; and as VALUES, value 1 is indicated. Thus, the macro computes horizontal sum (REDUPL=SUM) in *V1 – V4*.

Because *X1–X4* have empty cells, their corresponding cells in *V1 – V4* are tabooed and do not participate in the summation.

EXAMPLE 7. Researcher studies regularity and effects of family conflicts. She asked respondents to check, on what days of the past week (variables *MON – SUN*) quarrels began out of money, out of children, or out of relatives (respectively, codes 1, 2, 3 there). She also asked to rate, on the scale 1 to 4, the degree of psychological discomfort that is invoked by the arguments about these themes (variables *MONEY, KIDS, RELAT*). The researcher then wanted to know how many days must pass, minimum, from a quarrel to a quarrel on the same theme, in order for the respondent to rate their discomfort as such-and-such degree.

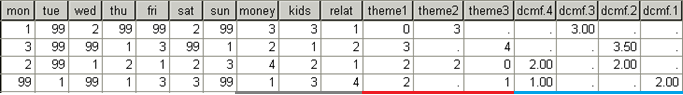
!KO\_valvars vars= mon tue wed thu fri sat sun /values= 1 2 3 /newvars= theme /newvals= ORDINAL

/redupl= MINGAP.

missing values theme1 theme2 theme3 (0).

!KO\_valvars vars= money kids relat /values= 4 3 2 1 /newvars= 'dcmf.' /newvals= theme1 theme2 theme3

/redupl= MEAN.



The first macro command ordered to create variables *THEME1 – THEME3*, in correspondence to the themes 1, 2, 3 contained in variables *MON – SUN*. She demanded that the new variables’ values be the minimal gap (redupl=MINGAP) in days (NEWVALS=ORDINAL, day’s ordinal number within week). Variables *THEME1, THEME2, THEME3* show “how many days, minimum, pass from a quarrel to a quarrel” on the theme, respectively, money, children, relatives. So, for the first respondent, arguments on the 1st theme had occurred just once, so gap=0, but on the 2nd theme arguments had been twice with interval 3 days (gap=3).

Because the researcher takes no interest in one-time per week quarrels, she set missing status to zero in the created variables, with MISSING VALUES command.

The second macro command ordered to create, out of the set *MONEY, KIDS, RELAT* with discomfort scores, variables *DCMF.4 – DCMF.1* corresponding to these 4 degrees of discomfort. These variables are the ones the researcher aimed to obtain for her future analysis. Their values, reporting “the number of days from a quarrel to a quarrel”, are borrowed from variables *THEME1 – THEME3*. For example, values in *DCMF.4* – are how many days pass from a quarrel to a quarrel about the same theme, which respondents consider discomforting to degree 4. Just in case in *MONEY, KIDS, RELAT* there encounters one and the same degree of discomfort more than once, it is requested to average (redupl=MEAN) the corresponding values taken from *THEME1 – THEME3*. So, the 2nd respondent had rated with 2 points both discomfort from the 1st theme and from the 3rd theme, therefore the values 3 and 4 (the intervals, in days) are averaged into 3.5.

***Special regimes***

The macro does not respond to weightedness, filteredness (FILTER, USE), splitness of the dataset. It doesn’t obey temporary (under TEMPORARY) transformations.

***Related macros***

!KO\_SRSREF (see below) with PACK=IREFER is equivalent to this macro when NEWVALS are variables. Macro !KO\_DERAND (“Derandomize tasks” collection) is able to give the same effect.

To count frequencies horizontally you may also by macros !KO\_HFREQ and !KO\_HCOUNT (“Horizontal tools” collection).

# MACRO !KO\_SRSREF: UNWRAPPING SERIES OF ITEMS OUT OF PACKED ENTRY MODE WITH THE HELP OF A REFERENCE VARIABLES SET

Version 1, Aug 2002. Tested on SPSS Statistics 11, 13, 26.

!KO\_srsref series= *s1 to s5* /\*Variables “series” containing data to be copied;

/\*should be consecutive, and write via “to”

/refer= *c1 c2 c3 c4* /\*Variables “reference” (MRC set) containing codes of the items selected;

/\*write either name-by-name or via “to”

/rated= *1 to 8* EXCEPT *3 5* /\*List of codes (non-negative numbers either name-by-name or range

/\*via “to”) of all items that were being rated; after the range may list, after

/\*keyword EXCEPT, codes that need be dropped from the range

/pack= MREFER /\*Mode of pack in the “series”: monotonic to the list RATED (MRATED),

/\*isomorphic (IREFER) or monotonic (MREFER) to the pack in REFER

/cap= *'x.'* /\*Prefix into varnames of a new series (SRS) being created

/rtimes= *1* /\*For PACK=IREFER, MREFER: how many times at max the same code could be selected

/\*by respondent (default=1)

/noans= *99* /\*Optional: value "no rating" into the new variables

/\*(an item was selected but its rating in SERIES is missing value)

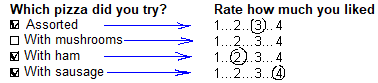
/nosel= *999* /\*Optional: value "item not selected" into the new variables,

/\*it will become user-missing.

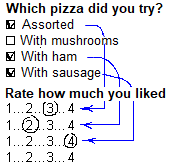
Minimal specification SERIES, REFER, RATED, PACK, CAP.

Imagine a question which asks to rate somehow (say, by a rating scale) items that were *first to be selected* in another question – a question of multiple response. A respondent rates not all items on the list, but only those items he had previously selected.

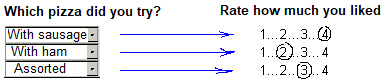
In a situation like this, if the speech is about manual data entry, there may be inconvenient to enter the data of rating into final, ready-to analyze variables, representing by themselves all the items on the list, - because that will call the data enterer to seek, in what of the variables she is to enter the response (the rating). Or it will call for programmic “skip/transit rules”, what isn’t always technically available. Instead of that, the responses-ratings can be entered by some *packed*, condensed mode, which would be more handy and quicker. Illustrations below clarify the idea.



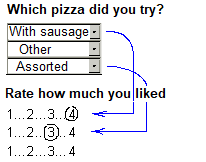
On the first scheme is the non-packed entry variant. For each item, in the order how they go in the questionnaire, there corresponds the variable – where the rating will be entered. Such variables are ready-for-analysis. Since item “With mushrooms” had not been previously chosen by the respondent, the corresponding variable-for-rating will be skipped during entry. This entry way can be called *isomorphic to item list*: each variable-for-rating corresponds to a strictly defined item.



On this scheme is the entry, *monotonic to item list*: items and variable-for-rating relate only by sequence. It is a packed (condense) entry, because only *first* three variable-for-rating will be engaged, and the 4th one will remain empty. Item “With mushrooms” was not selected; yet there might have been as well so that albeit it was selected, it is prohibited to be rated – by the questionnaire, - in both this and that cases the item is ignored when ratings are entered. In doing so, different respondents will come to have different number of ratings entered, with different shift of ratings towards the first variable-for-rating. However, for all respondents, item ratings must be entered in common sequence.



This type of entry is *isomorphic to item selection*. In the question to select items, a respondent could choose items in any order, and *in that* order his selected codes (items) will be entered in the categorical multiple response set, i.e., in the order “With sausage”–“With ham”–“Assorted”. Variables-for-rating strictly match the entered codes by position (index). Thus, item “Assorted” is selected and entered the 3rd; therefore, one needs to enter the rating for “Assorted” specifically in the 3rd variable-for-rating. Different respondents may have different entry order of items – and hence, of ratings.



This is the packed entry, *monotonic to item selection*, and differs from the previous entry by that variables-for rating relate to the entered codes (items) only by sequence. Items prohibited for rating are ignored while ratings are entered. Thus, item “Other”, the 2nd selected by the respondent, was not intended for rating, and so in the 2nd variable-for rating the rating of the next selected item, “Assorted”, will be entered. The 3rd variable-for rating will be left empty. As in the previous case, the entry order of items, and hence, of ratings, may differ for different respondents.

This macro is intended for unpacking “variables-with-ratings”[[6]](#footnote-6) entered in any of the latter three modes into the entry by the first mode, since only this entry is suitable for data processing. The macro takes (i) a series of variables condensely packed with ratings and (ii) a “reference” set of multiple response which contains codes of the items selected by respondents, and creates, from that series, new variables corresponding to all the items intended for being rated, by the questionnaire. This “unpacked” SRS series of variables is ready for analysis. The multiple response set serving the reference, the look-up about the items selected – must be categorical (MRC)[[7]](#footnote-7). The macro is suitable, too, for the case when an item had been permitted to select and rate more than once.

The macro does not check or clean the data. It just unpacks the data entered in a condensed manner.

***Subcommands***

**SERIES**

Input numeric variables with condensed (“leftward”, i.e., to the 1st variable) entry of “ratings”. The variables must be adjacent in the dataset, and you should write via “to”: *var* to *var*. Note that the packed entry does not mean compression due to selected but not rated, for refusion to rate reason, items. Rating of such items *should be present* in the input series – in the form of missing or the code designating “no answer”. The packing means compression due to items not selected by a respondent or prohibited for rating by the questionnaire.

In case of PACK=IREFER, the number of variables SERIES must equal the number of variables REFER. In either packing-of-ratings mode, the sequence of variables SERIES in the dataset must be without withdrawal. For example, if ratings were being entered condensely, left to right, into variables *VAR1 VAR2 VAR3*, you cannot cut out from the dataset or move to another position variable *VAR2* without damaging the correct unpacking of ratings.

**REFER**

Input numeric categorical variables containing (nonnegative) codes of items selected by respondents. They serve the reference about the selected items. It must be “categorical multiple response set” (MRC), i.e., several categorical variables with the same bank of response variants. You may write name-by-name or via “to”, but not mixed. In these data, there may be present both codes (items) permitted for rating and codes (items) prohibited for rating, if there are such. The latter are ignored by the macro.

Variables of the MRC set don’t have to bear “regular” build, i.e., the build whereby no missing is allowed to appear to the left of a valid code. The variable set needs not be registered in the data file as an MR set.

Single categorical variable may be indicated as REFER set.

**RATED**

The list of items permitted for rating – this number of SRS variables will be created. These must be nonnegative codes – values of REFER variables.

Note: it is the list of *all* the items potentially having been rated by the sample of respondents, and not of individual items which you might be interested in “presently” to unpack them into variables-for-analysis. The macro unwraps, into SRS series, not otherwise than all items which had been intended by the questionnaire for rating and which ratings, therefore, are stored in the variables-with ratings SERIES. If some item was permitted to rate – so ratings for it exist in SERIES, – you cannot leave its code out from the list RATED. Also, all the codes RATED must be valid values in REFER variables.

The list can be specified either by enumerating or by range via “to”: min to max; not mixed. In the first instance, the codes generally do not have to be integers. While if the codes are specified by range, the macro forms the list as integers from min to max (both these numbers must be integer).

By insertion of keyword EXCEPT after the range you can specify a list of codes (items) that you want to omit from the range (say, it can be items prohibited to rate: respondents did not rate them). For example, if VALUES= *1 to 5* EXCEPT *2 4*, then the list of codes the macro will generate is *1 3 5*. If no codes follow the word EXCEPT, then the word is ignored.

**PACK**

Indicate the type of condensed (i.e., rammed leftmost, to the 1st variable) entry in SERIES variables.

MRATED - entry monotonic to item list. Ratings are entered left to right in that *sequence* as items in the list RATED go. So should be for all respondents. The order of code entry in REFER variables plays no role.

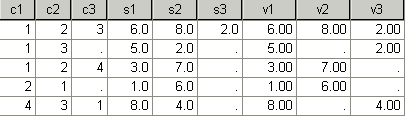
IREFER - entry isomorphic to item selection. The rating for an item is found in the SERIES variable of the *same* ordinal number (index) as the REFER variable containing that item code. Different respondents don’t have to have the same such corresponding order of codes and ratings. The number of SERIES variables must be equal to the number of REFER variables.

MREFER - entry monotonic to item selection. Ratings are entered left to right in that *sequence* as codes of their corresponding items are entered in REFER variables. Different respondents don’t have to have the same such corresponding order of codes and ratings.

With MRATED or MREFER entry mode, if fewer variables SERIES are input to the macro than there exist codes (items) permitted to rate, in reference variables REFER, ­­– then SPSS will issue warning: “The subscript in a vector reference is missing or otherwise invalid…” with indicating the case number where such shortage is detected. It means, the macro sought to take a rating from next variable of SERIES, but met no next variable. The said discrepancy between the number of codes in the reference set REFER and the number of variables-with-ratings SERIES can be the result of incorrectly entered data or incorrectly specified input, as well as the result of the intended input to the macro not all existing variables with packed entry of ratings, but rather only *first* so many (that latter is acceptable).

EXAMPLE 1. Pack monotonic to item list.

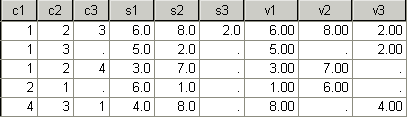
!KO\_srsref series= s1 to s3 /pack= MRATED /refer= c1 to c3 /rated= 1 2 3 /cap= v.



* “Variables-with-ratings” *S1 – S3* contain some scores or answers entered for all respondents in the sequence of rated items. These items (their codes) are specified in that sequence in RATED subcommand.
* MRC set *C1 – C3* contains the codes of items selected by respondents, and serves as a reference. Item coded 4, though it was being selected, was not intended to be rated afterwards (in the question, it meant “Other”), and so there can be no room for its ratings in *S1 – S3*; thus, it is not mentioned in RATED either.
* The macro unpacked the ratings into the new, ready-for-analysis variables *V1 V2 V3* (the codes entered into the variable names).

EXAMPLE 2. Same data entered by another mode: pack monotonic to item selection.

!KO\_srsref series= s1 to s3 /pack= MREFER /refer= c1 to c3 /rated= 1 2 3 /cap= v.



* Entry in “variables-with-ratings” *S1 – S3* sequentially corresponds to code entry in the reference set *C1 – C3*. (It becomes clear, if to compare, with the previous example, how ratings of the 2 last respondents are entered.) The order of listing codes in RATED is of no importance; it only defines the queue of output variables *V1 – V3*.
* Results of unpacking are the same as in the previous example.

**CAP**

Specify prefix into the names of output variables, You may take it in quotes or apostrophes (they are necessary if you end the prefix with a period as a separator: ‘var.’). The variable names will be formed of the prefix and the codes RATED. The new variables of these name must not exist yet in the dataset.

**RTIMES**

This subcommand is for the case if items permitted for rating could be, as the questionnaire stipulates, selected and rated more than once. For example, a multiple response question could ask, “TV sets of which manufacturers are in your family?”, and allow to name the same manufacturer several times, if there are several TV sets of one the same manufacturer in the family. That means, the same code may repeat in REFER variables on the rights on independent items, each of which will be rated in SERIES variables by its own rating.

Specify here how many times maximally could an item from RATED be selected/rated. Commonly, this is the number of REFER variables itself, but you may specify a number less than that: it this case, ratings only for the first RTIMES encounters of a code will be transferred to the being created variables. The macro creates as many variables for every item, as RTIMES is set to, and the names of the output variables end with indices designating “1st encounter” “2nd encounter», and so forth.

By default/unspecifying of the subcommand, RTIMES=1, i.e., no code duplicating in REFER variables is provided (should be no repetitions). RTIMES *is inactive* with PACK=MRATED: with this entry type, the macro does not react to repetitions of codes in REFER, it cares only if the code is found there or not.

**NOANS, NOSEL**

By default/unspecifying these subcommands the macro leaves, in the output SRS variables, empty cells, where it inserted no valid rating. You can request what values (any numbers) to put there.

NOANS means “no answer”, that is, the item permitted for rating was selected, but its rating in the SERIES variables appeared to be missing (system- or user-missing). NOANS value will not be made missing by the macro.

NOSEL means “item not selected”, that is, no code of the item was found in REFER. NOSEL value becomes user-missing in the output variables. It is acceptable to specify one the same value for NOANS and NOSEL.

In any way, the macro leaves, in all the output SRS series, those cases empty (system-missing) that were missing (system- or user-) in *all* the variables of the reference set REFER; in other words, it is respondents who were not asked to select items at all.

EXAMPLE 3.

!KO\_srsref series= i1 to i14 /pack= MREFER /refer= mrc1 to mrc8 /rated= 1 to 20 EXCEPT 19 /cap= 'v.'

/rtimes= 3 /noans= 99 /nosel= 999.

* Entry in variables-with-ratings *I1 – I14* sequentially corresponds to code entry in the reference set *MRC1 – MRC8*. In SERIES, there are packed ratings of product brands which are coded in REFER with codes 1–20 (except brand 19, which was not intended for rating).
* Respondent could select and rate the same brand up to three times (RTIMES=3).
* In the output unpacked variables-with-ratings (their names start with *V.*) absence of the rating for a selected code will be coded as 99, whereas if the rating is missing because the brand was not selected – then as 999 (and this code will be user-missing).

***Special regimes***

The macro does not respond to weightedness, filteredness (FILTER, USE), splitness of the dataset. It doesn’t obey temporary (under TEMPORARY) transformations.

***Related macros***

If you want to “unwrap” into items not simply a matrix question but heterogeneous variables, and entry in them follows the logic of IREFER entry, use macro !KO\_DERAND (collection “Derandomize tasks”).

1. If a value is encountered more than once, only the last its encounter will count. [↑](#footnote-ref-1)
2. Which you can specify directly or indicate the variables containing them; in the latter case different new values can be for different cases. [↑](#footnote-ref-2)
3. Negative values – you will have to recode them in advance into nonnegative, in the Recode menu. [↑](#footnote-ref-3)
4. If such variables are already in the dataset, SPSS will issue an error message, however the macro will continue its work and will affect those variables. So be attentive. [↑](#footnote-ref-4)
5. You can manage random numbers generation seed in Random Number Generators menu. [↑](#footnote-ref-5)
6. Quoted, because actually various data can be in these variables, not only rating scale scores. The word “rating” is utilized provisionally in the description of this macro. [↑](#footnote-ref-6)
7. If it is dichotomous (MRD), re-do it first into categorical by macro !KO\_MRDMRC (collection “Categorical – Binary recodings”). Only the variant with packing monotonic to items list, MRATED, will be feasible. [↑](#footnote-ref-7)