***Various proximities***

SPSS macros by Kirill Orlov

kior@akado.ru, ttnphns@gmail.com

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

All rights reserved

*Various proximity measures.* Calculation of a large number of measures of proximity or association (similarities, distances, correlations) many of which are absent in SPSS. Among them is Gower similarity for comparing respondents by quantitative and qualitative characteristics at once.

*Read “*[*About SPSS macros*](https://www.spsstools.net/en/KO-aboutmacros)*” what are they and how to run them.*

*The “Protected directory” error.* Some of the macros described in the current document write temporary files to hard disc. If you don't have full Administrator rights of your computer, it may cause error saying, among things: *“SPSS Statistics cannot access a file... specifies a protected directory...”*, meaning that the default directory the macro wants to use is protected on your PC. To solve the problem, in Syntax window issue command: CD 'myfolder'., where 'myfolder' is the path/name of some folder where you are allowed to save files to.

* [!KO\_EDPROXMX](#_MACRO_!EDPROXMX:_EDITING) is an auxiliary macro allowing to do some editing job on an existing square matrix of proximitiy, for example extract a submatrix from it, or make a matrix symmetric, change its diagonal, etc.
* Gower similarity [!KO\_GOWER](#_MACRO_!GOWER:_GOWER_1) to compute likeness between cases on quantitative and qualitative characteristics at once.
* Various proximities – distances, similarities – for quantitative data [!KO\_PROXQNT](#_MACRO_!CANBER:_CANBERRA).
* Various proximities – distances, similarities – for binary data [!KO\_PROXBIN](#_MACRO_!PROXBIN:_VARIOUS).
* Tetrachoric correlation coefficient [!KO\_TETRACH](#_MACRO_!TETRACH:_TETRACHORIC_1).
* Biserial correlation coefficient [!KO\_BISER](#_MACRO_!BISER:_BISERIAL).
* Autocorrelative distance [!KO\_ACORRD](#_MACRO_!ACORRD:_AUTOCORRELATIVE).
* Rescaled Pearson correlation coefficient [!KO\_RESCR](#_MACRO_!RESCR:_RESCALED).

*Attention*, for all these macros it is recommended that names of input variables be no longer than 8 bytes, otherwise they will be truncated at output.

# MACRO !KO\_EDPROXMX: EDITING OF A PROXIMITY MATRIX

Version 2, Sep 2023 (Version 1, Mar 2020). Tested on SPSS Statistics 22, 26, 28.

!KO\_edproxmx matrix= *var1 to var250* /\*Columns forming the matrix body; may use “to”

/seq= ROW /\*Sequence of rows/columns in the output matrix: (COL, default)

/\*ROW, GRAD1, GRAD2, IND *'file'*

/oper= AVER /\*Operation in the body: NONE (тж п/у), AVER, GMEAN, HMEAN, UBYL, LBYU,

/\*SWITHG, GWITHS, FLIP, DIAG *v*, ABS, RANK, URANK, LRANK,

/\*UADD v, LADD v, UMULT v, LMULT v, UEXP v, LEXP v

/numvars= *height bmi* /\*Optional: take along these numeric variables

/strvars= *v3 to v5* /\*Optional: take along these short string variables

/dgelem= /\*With seq=GRAD1/GRAD2: diag elements participate (YES, default) or not (NO)

/savedg= /\*Optional: save the diagonal as an external file (path/filename or dataset)

/savetr= /\*Optional: unwrap ans save the triabgles as an external file

/\*(path/filename or dataset)

/plot= GREY /\*Produce heatmap: NONE (default), COLOR or GREY or RGREY;

/\*word LABEL can be appended then.

/bounds= OBS /\*If PLOT specified: bounds for the heat scale on the map: AUTO (default),

/\*OBS, or min max.

Minimal specification MATRIX.

This macro does not compute proximities or statistics, rather, it serves an editing tool for an existing proximity matrix between items of one set – i.e. “square matrix” (corresponds to a unipartite graph). The macro can do the following things:

* Compile the matrix in any you want regular order – that is, so that items defining rows/columns go in its rows and its columns in one same sequence – *co-orded* (consequently, elements *i,i* will lie on the diagonal).
* Extract a submatrix.
* Do some operations inside the matrix body (for example, make asymmetric matrix symmetric, add a constant, transpose, set a diagonal).
* Save out the diagonal or the unwrapped triangles of the matrix.
* Draw a heatmap of the matrix.

The macro accepts the matrix, does its job and outputs the obtained matrix in a new unnamed dataset. The macro does not create variable *ROWTYPE\_* for the matrix. You can create it manually or borrow it from the initial dataset (see s/c STRVARS).

EXAMPLE 1. Extraction and reordering.

proximities v1 to v100 /view= case /measure= euclid /matrix= out(\*) /print= none.

dataset name prox.

!KO\_edproxmx matrix= var15 var14 var10 var20 to var24 /numvars= caseno\_ /strvars= rowtype\_.

value label rowtype\_ 'PROX' 'DISSIMILARITY EUCLID'.

* PROXIMITIES command created a square distance matrix between the data cases and output it as a new dataset.
* Macro !KO\_EDPROXMX took only the following rows/columns of it and in this order: 15 14 10 20 to 24. And saved this matrix as a new unnamed dataset.
* Also, the macro took up, and saved, variables of the initial dataset: CASENO\_ and ROWTYPE\_ (they were subsidiary to the matrix). The instructing value label in ROWTYPE\_ was restored by VALUE LABELS command.

EXAMPLE 2. Simply draw heatmap for a correlation matrix.

dataset declare corr.

correlations /variables= v1 to v10 /matrix= out(corr).

dataset activate corrmat.

select if rowtype\_='CORR'.

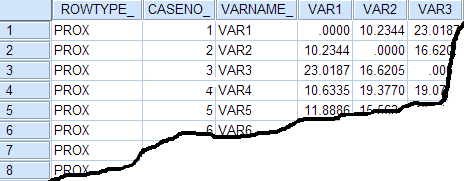
!KO\_edproxmx matrix= v1 to v10 /plot= COLOR.

* CORRELATIONS command created a square correlation matrix between the data variables and output it as the dataset *CORR*.
* That dataset is activated and only rows corresponding to the correlations are selected to remain in the matrix.
* !KO\_EDPROXMX took the complete matrix body and plotted the heatmap.

***Input matrix***

The dataset must contain a “matrix of pairwise proximities” between items of one set. Generally, there can be any numeric values in the matrix, but for the missings. The matrix can be symmetric or asymmetric. Variable names – matrix columns – up to 8 bytes. Required is the presence of short string variable *VARNAME\_* naming rows in correspondence to columns. Names, which are the values of the variable, should be written in the same case of letters as the identical to them names among the columns. The macro does *not* require that the rows and the columns go in the same order or their number and list be fully identical: it is just the task of the macro which is to select from the input matrix the rows and columns same by their names, and to co-order them – to return you the matrix of square shape and co-ordered (diagonalized) built. Variable *ROWTYPE\_* and other auxiliary are not necessary in the input matrix.

Apart from the variables defining the matrix body and the *VARNAME\_* variable, the dataset may contain any other variables.



***Subcommands***

**MATRIX**

Specify variables of the working dataset which are proper the columns of the matrix of proximities. You may list all or just needed columns and in arbitrary sequence. May use “to” to specify by range.

*Specification of open range with the help of “?”*. In some instances there may arise need to specify a range between a pair of variables which themselves are not included in the range. Use “?” at the sides, for that. For example, *?VARNAME\_ to ENDVAR?* means all variables found in the dataset between variables *VARNAME\_* and *ENDVAR*, excluding these two. To specify a range which is open from one side, use “?” only from that side. For example: *?VARNAME\_ to VAR100* or *VAR1 to ENDVAR?*.

When you want the macro to extract a square submatrix from a bigger matrix you can go two ways: (1) specify only the columns you need – this is done in MATRIX s/c; and/or (2) select only the rows you need – this is done by selecting or filtering of cases of the dataset. In any way the macro will gather the square and co-ordered (sub)matrix, because there exists variable *VARNAME\_*, storing the names of items, and there are column names – all about the same items.

**SEQ**

This subcommand sets in what sequence items (i.e. rows/columns) must go in the output matrix:

COL - (also default/unspecification) rows and columns will go in the order of column names list according to the MATRIX subcommand.

ROW - rows and columns will go in the order of rows (values of VARNAME\_) in the input matrix. If the matrix, as input by MATRIX subcommand, is already square and with co-ordered rows and columns, this will yield the same result as SEQ=COL.

GRAD1 - do sorting of rows/columns by “multiplicative grade”. For each pair of *same-name* row *R* and column *C* of the matrix sized *I×I*, ready to output, there is computed value , i.e. the crossproduct. Then rows/columns get sorted by the decrease of this magnitude.

GRAD2 - do sorting of rows/columns by “additive grade”. For each pair of *same-name* row *R* and column *C* of the matrix sized *I×I*, ready to output, there is computed value , i.e. the combined sum. Then rows/columns get sorted by the decrease of this magnitude.

IND *file* - sort by indices stored as an external .SAV file. After the keyword, specify the path/name of the file in quotes or apostrophes. The file must be of a single variable (of any name) which values are the indices (positions) of items in the output matrix. For instance, if the 8th case in the *file* is value 17, then the 8th row/column of the input matrix will become the 17th in the output matrix. Indices should be positive integers (fractional values will be truncated by the macro) not greater than the number of rows/columns in the matrix.

For GRAD1/GRAD2, subcommand DGELEM is significant. Also to remark: reordering GRAD1/GRAD2 depends on values in the matrix body, and it is done *after* optional operation with elements (s/c OPER).

*Remarks on* IND.

1. This option demands the input matrix (the matrix as it is taken by subcommand MATRIX) to be already square and with co-ordered rows/columns. Else the macro will issue an error message.
2. Indices in *file* may repeat; the number of cases therein may exceed or be less than the number of items in the matrix taken by MATRIX subcommand. Thus, SEQ=IND allows not only to set the sequence of rows/columns, but also to extract submatrix from matrix or, on the counter, to propagate some rows/columns whatever times one wishes.

EXAMPLE 3. Change items order in matrix for another, random.

compute sortvar= uniform(1).

sort cases by sortvar.

!KO\_edproxmx matrix= var1 to var100 /seq= ROW.

* Rows of the input matrix were sorted by values of random variable SORTVAR.
* SEQ=ROW of the macro compiles the matrix with the order of rows/columns following the order of rows of the input matrix.

**OPER**

Optionally you can request one operation to be done over elements of the matrix body. Choose:

NONE - (also default/unspecification) do nothing.

AVER - make matrix symmetric by averaging (usual mean) its symmetric-positioned elements.

GMEAN - make matrix symmetric by averaging (geometric mean) its symmetric-positioned elements; requires nonnegative values.

HMEAN - make matrix symmetric by averaging (harmonic mean) its symmetric-positioned elements; requires nonnegative values.

UBYL - make matrix symmetric by replacing an element of the upper triangle with its symmetric-positioned element from the lower triangle.

LBYU - make matrix symmetric by replacing an element of the lower triangle with its symmetric-positioned element from the upper triangle.

SWITHG - make matrix symmetric by replacing the lesser element by the greater element, in a pair of symmetric-positioned elements.

GWITHS - make matrix symmetric by replacing the greater element by the lesser element, in a pair of symmetric-positioned elements.

FLIP - transpose, i.e. swap the upper and the lower triangles by their places.

ABS - abolish negative sign of off-diagonal elements.

RANK - rank ascendingly off-diagonal elements.

URANK - rank ascendingly elements of the upper triangle.

LRANK - rank ascendingly elements of the lower triangle.

DIAG *val* or *var* - set new diagonal to the matrix. After the keyword, specify either a scalar (number) or a name of a numeric variable in the dataset; values from this variable will be placed on the diagonal; missings are not allowed in the variable.

UADD *val* - add a constant to elements of the upper triangle. Specify a number after the keyword.

LADD *val* - add a constant to elements of the lower triangle. Specify a number after the keyword.

UMULT *val* - multiply elements of the upper triangle by a constant. Specify a number after the keyword.

LMULT *val* - multiply elements of the lower triangle by a constant. Specify a number after the keyword.

UEXP *val* - exponentiate elements of the upper triangle: raise to constant power. Specify a nonzero number (power) after the keyword.

LEXP *val* - exponentiate elements of the lower triangle: raise to constant power. Specify a nonzero number (power) after the keyword.

*Remarks on* UEXP/LEXP.

1. If there are negative values in the triangle, the exponent (power) must be integer.
2. If there are zero values in the triangle, the exponent (power) must be positive.

EXAMPLE 4.

!KO\_edproxmx matrix= var1 to var100 /seq= GRAD1 /DGELEM= YES /OPER= DIAG newdiag /plot= COLOR.

* Columns VAR1 thru VAR100 of the proximity matrix are taken (they could be the whole matrix or maybe just first 100 its columns).
* SEQ=GRAD1 requests to reorder the rows/columns by elements’ magnitude, “multiplicitly”, and contributions of the diagonal elements should be accounted at that: DGELEM=YES.
* OPER=DIAG sets new diagonal which values are borrowed from variable *NEWDIAG* of the input dataset. SEQ=GRAD1 reorders after OPER, so accounted are new, not old, diagonal elements.

**NUMVARS, STRVARS**

You can take along some variables of the dataset, extraneous to the matrix body, so that their cases (the rows) will undergo reordering or selection following the compilation of the proximity matrix. Specify numeric variables in NUMVARS subcommand and string variables in STRVARS subcommand. May use “to” to specify by range. Names – up to 8 bytes long. String variables must be short (variable width up to 8 bytes). Value: -999 is not permitted in the numeric variables. Do *not* enter variable *VARNAME\_*: the macro itself creates this variable for the output.

Missings are allowed in the variables, however user-missing values will lose its missing status at the output to a new dataset. The status, as well as variable and value labels – they won’t be preserved by the macro – you will have to restore yourself (it is easy to do via menu Data – Copy Data Properties).

**DGELEM**

This subcommand is needed only for SEQ= DIAG1 or DIAG2. Set DGELEM=YES (also default) if you want diagonal entries of the matrix (i.e. the values on the crossing of the same-name rows and columns) to participate in computation of the “grade”, by descend of which rows/columns will be sorted. Set DGELEM=NO if diagonal entries should be ignored at computation of the “grade”.

**SAVEDG**

By this subcommand you can save the diagonal of the matrix, ready for output, as an external .SAV file or a dataset. Specify path/name of the file in quotes or apostrophes or the name of a declared dataset. Variable *VARNAME\_* is saved too.

**SAVETR**

By this subcommand you can save the unwrapped triangles of the matrix, ready for output, as an external .SAV file or a dataset. Specify path/name of the file in quotes or apostrophes or the name of a declared dataset. The following columns will be saved. *LOWER* (lower triangle), *UPPER* (upper triangle); a case is a pair of positionally symmetric elements of the matrix. Columns *COLROW* and *ELEM* show the location, coordinates of those elements in the matrix: *COLROW* is the column number (for a lower element, *LOWER*) or the row number (for an upper element, *UPPER*), and *ELEM* is the element number in that column or row. Columns *VARNAME1* and *VARNAME2* are the corresponding names from *VARNAME\_*.

**PLOT**

Draws heatmap of the resultant, output matrix. PLOT=COLOR makes it colour (magnitude of elements is rendered by hue). PLOT= GREY or RGREY makes it greyscale (magnitude of elements is rendered by brightness, under GREY, or opposite, by darkness, under RGREY). Next you may append the second keyword LABEL – to label the cells with values.

PLOT=NONE (also default) does not produce the graph.

**BOUNDS**

This subcommand is in effect only with PLOT specified. It sets bounds for the colour (or brightness) scale.

AUTO - (also default/unspecification) let SPSS automatically define suitable bounds.

OBS - the bounds exactly match with the observed minimal and maximal values in the matrix.

*min* *max* - specify the bounds manually as two numbers, minimum and maximum. Indicate values by and large comparable with the values of the matrix.

Manual specification of the bounds means that you are fixing the colour or brightness mapping on the picture with respect to the size of the elements in the matrix. It becomes possible to compare different matrices with each other. For example, if you specify *min* *max* as *-1 1*, then a correlation matrix comprised of lesser coefficients by absolute magnitude will appear more dim or more monochrome than a matrix comprised of greater coefficients by absolute magnitude; so such pictures “on the same scale” can be compared.

***Special regimes***

The macro ignores weighting of the dataset (however, it doesn’t take in the procedure cases with missing and nonpositive weights) and is not suited for splitting the dataset. The macro obeys case selection/filtering (commands FILTER, USE, SELECT IF, N OF CASES) and temporary (under TEMPORARY) transformations.

# MACRO !KO\_GOWER: GOWER SIMILARITY

Version 5, Oct 2019 (Version 1, Dec 2000). Tested on SPSS Statistics 20, 22, 25.

!KO\_gower sca= *v1 v2 v10* /\*Scale variables, if any, name-by-name list

/ord= *v3 v5* /\*Ordinal variables, if any, name-by-name list

/bin= *v4 v8* /\*Binary variables (0 vs 1), if any, name-by-name list

/nom= *v6* /\*Nominal variables (numeric), if any, name-by-name list

/cnt= *v7 v16* /\*Count variables, if any, name-by-name list

/weights= *1 1 1.5 1 1 1 1 2 1 0.5* /\*Optional: weights for input variables,

/\*in sequence SCA ORD BIN NOM CNT: some nonnegative numbers

/missing= /\*Treatment of missing values: LISTWISE (default) - delete cases

/\*listwise; PAIRWISE – take all cases, using in calculations all valid data

/\*whenever possible. Optional additional second keyword

/\*ASVALID – consider user-missing data as valid

/id= /\*Optional: string case-identifying variable

/flatten= /\*Optional, for scale variables: flatten kurtosis in these variables: name-by-name list

/add05= /\*Optional, for count variables: add 0.5 to counts in these variables: name-by-name list

/reverse= /\*Reverse Gower similarity into distance: don’t do (NONE, default),

/\*subtract from 1 (ONEMINUS), sq. root after subtraction from 1 (RONEMINUS)

/\*angle in radians (ARCCOS)

/ecorrect= /\*Check and correct distances for euclidity: don’t do (NONE, default),

/\*LINGOES (Lingoes method), CAILLIEZ (Cailliez method).

Minimal specification at least one of SCA, ORD, BIN, NOM, CNT.

Gower similarity coefficient (Gower, 1971) is a measure of likeness between two objects (namely, cases, not variables). Its feature is that it is able to take into account together characteristics that are heterogeneous by their measuring nature. Some of the variables on base of which this summarizing index of similarity establishes itself may be quantitative or ordinal, others may be binary (characteristic either present or absent), still other may be nominal (alternative categories, such as sex or profession). Gower offered coefficient for a mixture of scale, binary and nominal variables. Summonds for ordinal and count variables were added later.

The coefficient of similarity between objects *i* and *j* is the weighted average of their similarities computed for each of the *p* input variables (*p*≥1):

,

where *k* is a variable, *Sijk* is the similarity by it, *wk* is the weight appointed to that variable. The weights can be supplied; by default, they all are 1. The partial similarity *Sijk* varies in the range [0,1] and is computed differently depending on the variable *k* type:

* If the variable is **nominal**, then *Sijk* = 1, if *xik* =*xjk*, i.e. the values of objects *i* and *j* by variable *k* are same; otherwise *Sijk* = 0. Thus, if all *p* variables are nominal, Gower coefficient is equal to **Dice** matching coefficient which one obtains if he recodes the nominal variables into binary dummy ones.
* If the variable is **binary** (1 vs 0), then *Sijk* = 1, if *xik* =*xjk* = 1; otherwise *Sijk* = 0. Also, if *xik* =*xjk* = 0, then at the comparison of *i* with *j* by variable *k* the weight zeroes off: *wk* = 0. Hence, if all *p* variables are binary, Gower coefficient turns into **Jaccard** matching coefficient.
* If the variable is **scale** (interval or ratio), then

,

where *xk\_max* and *xk\_min* are, respectively, the maximum and the minimum in variable *k* (if they are equal, *Sijk*  is taken for 1). Thus, if all *p* variables are scale, Gower coefficient is the **range-normalized Manhattan distance** reversed into similarity.

* If the variable is **ordinal**, then according to Podani’s innovation (Podani, 1999), variable *k* is ranked (*x* get replaced by ranks *r*) classically (i.e. with averaging of ties), and then

*Sijk* = 1, if *rik* = *rjk*;

otherwise:

,

where *rk\_max* and *rk\_min* are, respectively, the maximal and the minimal ranks in variable *k*; *Tk\_max* and *Tk\_min* are the number of objects with, respectively, the maximal and the minimal rank; *Tik* and *Tjk* are the number of objects with the same rank as, respectively, objects’ *i* and *j* (including *i* and *j* themselves). This formula specially levels out “distancing” intervention of tied ranks. Podani offered also a simpler variant without such levelling: after ranking the variable, use with the ranks the same formula as is used for a scale variable.

* If the variable is **count** (frequencies), then, by the addition of the macro’s author, *xik* and*xjk* are considered as a pair of complementary frequencies which are subject to comparison by the chi-square of agreement, that is

,

where the expected frequency is the average of the two being compared: ,

so that , from where the partial similarity comes out as:

; and if both frequencies are zero, *Sijk* = 1.

Thus, if all *p* variables are count, Gower similarity is **Canberra distance** reversed into similarity.

Sources:

* Gower, J. C. A general coefficient of similarity and some of its properties // *Biometrics*, 1971, 27, 857-872.
* Podani, J. Extending Gower’s general coefficient of similarity to ordinal characters // *Taxon*, 1999, 48, 331-340.

The macro outputs square symmetric matrix of Gower coefficients as new unnamed working dataset. There is an option to give weights to the input variables, to manage contribution of each variable into coefficient. If cases with missing values were taken (MISSING=PAIRWISE), then cases are being compared pairwise by those variables only where they both are valid (besides that, in binary variables – as said before – comparison is made only when at least one of the two cases owns the attribute). If happens that – due to missings - for some cases there is not enough data to compare them with other cases, then the macro will exclude these cases from the final matrix of coefficients and will notify.

Column *CASENO\_* by the matrix is the case number in the input dataset.

Input variables *must not contain value: -999*. The working dataset should have no variables with names CASENO#$ or STRID#$.

Since partial similarities computed by scale variables get normalized by their observed ranges, removing some cases from dataset by the user may potentially affect coefficients that will be computed for the remaining cases.

Like majority of similarity coefficients, Gower coefficient takes values from 0 to 1. You can reverse similarity into distance by s/c REVERSE.

***Subcommands***

**SCA**

Name-by-name input scale variables. It could be interval or ratio level of measurement. Measurement units and magnitudes may be different in the variables – it does not affect result, because similarities by each variable are normalized via division by the variable’s range. You should, however, watch after outliers because they enlarge range.

**ORD**

Name-by-name input ordinal variables (numeric) - the ones you treat as ordered categorical measurement scale. Measurement units and magnitudes may be different in the variables – it does not affect result, because the macro ranks values in these variables.

As a variant, you may choose not to use ORD subcommand but instead replace the ordinal variables by their rank variables (menu Transform – Rank Cases) and process those as scale ones, specifying in SCA subcommand. However, in the presence of much amount of tied ranks (ties) this variant should be acknowledged less correct.

**BIN**

Name-by-name input numeric binary variables. Variables of type “attribute present – attribute absent” are meant here, not nominal dichotomous variables (like “female – male”). Attribute present must be coded as 1 and absent must be coded as 0.

**NOM**

Name-by-name input numeric nominal variables. Category coding may be arbitrary.

**CNT**

Name-by-name input numeric frequency count variables. Values are nonnegative integers.

**WEIGHTS**

Optional subcommand where you may attach weights to the input variables – their contribution to the overall computed coefficient. These may be any nonnegative numbers in ratios between them that you need (at least some must be positive). Only ratios of magnitudes, not magnitudes themselves, play role. There must be as many values as there are input variables, and they must go in the variables sequence following order SCA ORD BIN NOM CNT.

EXAMPLE 1.

!KO\_gower sca= v1 v2 v3 /bin= /nom= w6 occup /weights= 2 2 2 3 3.

* Variables V1 V2 V3 W6 OCCUP are given, respectively, weights 2 2 2 3 3.
* There is no ordinal variables: s/c ORD omitted. Binary variables are also absent, s/c BIN might have been omitted either.

**FLATTEN**

This option concerns only scale variables (SCA). You can list name-by-name those variables of the specified in SCA, in which you want to flatten the shape of distribution. FLATTEN squeezes, prior computing similarity, distribution in a variable from “above”, thus lessening kurtosis and shortening long “tails”. You may want to use option FLATTEN only for those continuous variables with the kurtosis much bigger than of the normal distribution. In such a sharp-top variable, the range is too wide in comparison with differences between the majority of its observations, and so *Sijk* in bulk approaches 1, – if FLATTEN is not done. FLATTEN flattens distribution by transforming the variable via formula , where *Z* is the variable after its standardization.

EXAMPLE 2.

!KO\_gower sca= v1 v2 v3 /bin= b2 b7 b8 b13 /nom= region /flatten= v2.

* Continuous variable V2 has sharp peak and long tails, therefore it was decided to flatten its distribution.

**ADD05**

This option concerns only count variables (CNT). You can list name-by-name those variables of the specified in CNT, in which you want to add 0.5 to values, in order to get rid of zeroes. ADD05 adds, prior computing similarity, magnitude 0.5 to all frequencies. The similarity between any frequency and zero frequency always equals 0. For example, 1 - |9-0| / (9+0) is zero similarity, as well as in 1 - |1-0| / (1+0), in spite of that 9-0 is a much bigger difference than 1-0. If you don’t like this fact (although generally it doesn’t lack sense for frequencies), then request ADD05 for all the CNT variables or only for those of them where you want to get rid of zeroes. Then, 1 - |9.5-0.5| / (9.5+0.5) = 0.1, which will be less similarity than 1 - |1.5-0.5| / (1.5+0.5) = 0.5.

**MISSING**

Subcommand to treat missing data. Select:

LISTWISE – (also default/unspecification) listwise deletion of cases with missings. If a case has at least one missing value in the variables analyzed by the macro the case is omitted from the analysis.

PAIRWISE – admit all cases to analysis. Comparison in a pair of cases takes place only by those variables where both values are valid.

After either of the two keywords you may optionally write ASVALID, if you want the macro to perceive user-missing values, be those in the analyzed variables, as valid values. By default, the macro treats user-missing values as missing.

**ID**

You may specify string variable (up to 8 bytes wide) to serve as case identifier in the output matrix of coefficients. By default/unspecification of the subcommand the macro will create string identifier named *ID*, which values are Latin “C” plus the case number coinciding with *CASENO\_*.

**REVERSE**

Gower measure *S* is a similarity. By default and by REVERSE=NONE, the macro returns a matrix of the similarities. You may want to turn similarity into “Gower distance” *D* by any of the following formulas:

ONEMINUS *D* = 1-*S*. This distance varies from 0 to 1, like *S*.

RONEMINUS *D* = sqrt(1-*S*). This distance varies from 0 to 1, like *S*.

ARCCOS *D* = arccos(*S*). This distance sees *S* as the cosine of an angle, and it itself is that angle in radians.

*Properties of these distances*

Metricity (satisfaction of the triangular inequality):

All three kinds of distances, 1-*S*, sqrt(1-*S*), and arccos(*S*) are metrics, if s/c ORD was not specified. With ORD specified, they may no longer be metrics (may violate “triangular inequality”).

Geometrical euclidity (convergence of distances in Euclidean space):

sqrt(1-*S*) are guaranteed euclidean if s/c ORD was not specified. With ORD specified, they may no longer be euclidean.

arccos(*S*) are guaranteed euclidean if only s/c BIN was specified.

1-*S* are almost always non-euclidean.

You can request to treat the obtained matrix of distances until it is euclidean by subcommand ECORRECT. If distances are geometrically euclidean they are automatically metric too.

**ECORRECT**

This optional subcommand acts if REVERSE≠NONE, else it is ignored. Not seldom it is desirable for a user that the observed (computed) distances converge in Euclidean space (geometrical euclidity): then every analysis suited for genuine euclidean distances can be done with these distances too. Above, in s/c REVERSE it is written which of the three kinds of “Gower distance” appear to be geometrically euclidean and when.

If you got matrix (sized not less than 3) where distances might not converge in Euclidean space, and you want to make them to converge, take advantage of subcommand ECORRECT. This subcommand adds, to all distances of the matrix, a minimal constant sufficient to make them euclidean. One should not forget though, that adding a constant is a distortion of the original proximities; therefore it is desirable that the constant be small. The macro will inform of the magnitude of the constant needed. Select the method of the constant computation:

NONE - (also default/unspecification) don’t test the distances and don’t correct them to euclidity.

LINGOES - Lingoes method. Computed is the constant to add to the squares of distances, nonsquared distances to become euclidean. It is a faster method, but distances in the matrix grow unevenly (because their *squares* get modified).

CAILLIEZ - Cailliez method. Computed is the constant to add to distances as they are (nonsquared), and they become euclidean. This method not infrequently lead to a greater inflation of distances than the previous method does, but it modifies distances in matrix evenly.

If the distance matrix does not need correction because it is euclidean already, the macro will not calculate the constant and will inform of that.

Lingoes constant is established as the quantity slightly greater than , where is the last (minimal) eigenvalue of matrix **C**, which is the so called *double centrate* of the distance matrix *D* (returned by REVERSE subcommand) square-exponentiated. (The double centrate is the distance matrix converted to the matrix of scalar products.)

Cailliez constant is established as the quantity slightly greater than the first (maximal) eigenvalue of the following asymmetric matrix:

, where **P** is the double centrate of the distance matrix, obtained the same way as was **C** but without exponentiation of *D* to their squares.

Correction is successful if the double centration matrix (the double centrate) **C** lost its negative eigenvalues.

Correction onto euclidity may be calculatory expensive with a big matrix and demand time. Therefore it is not recommended to resort to ECORRECT, especially CAILLIEZ method, when the size of the matrix, i.e. the number of cases, approaches to or exceeds 1000.

EXAMPLE 3.

!KO\_gower ord= item3 item4 item5 item6 /bin= b1 b2 b3 b4 b5 /nom= n9 /reverse= ONEMINUS /ecorrect= LINGOES.

* REVERSE ordered to reverse Gower similarity into dissimilarity by formula 1-S.
* Likely this distance won’t support Euclidean space, so requested is Lingoes correction to euclidity.

Sources:

* Gower, G.C., Legendre, P. Metric and euclidean properties of dissimilarity coefficients // *Journal of Classification*, 1986, 3, 5-48.
* Choi, H., Choi, S. Robust kernel Isomap // *Pattern Recognition*, 2007, 40, 853–862.

***Special regimes***

The macro ignores weighting of the dataset (however, it doesn’t take in the procedure cases with missing and nonpositive weights) and is not suited for splitting the dataset. The macro obeys case selection/filtering (commands FILTER, USE, SELECT IF, N OF CASES) and temporary (under TEMPORARY) transformations. If you use selection by commands SELECT IF or N OF CASES not under TEMPORARY, you’ve removed a portion of cases from the original dataset and the values of column *CASENO\_* by output matrix are then the case numbers in that remaining dataset. In all other screening regimes, the values of *CASENO\_* are the case numbers in the original dataset observed there prior your screening of cases.

# MACRO !KO\_PROXQNT: VARIOUS PROXIMITY MEASURES FOR QUANTITATIVE DATA

Version 2, Sep 2020 (Version 1, Feb 2018). Tested on SPSS Statistics 20, 22, 25.

!KO\_proxqnt vars= *v1 to v15* /\*List of numeric variables (may use “to”)

/view= CASE /\*Compute proximities between cases (CASE, default) or

/\*between variables (VARIABLE)

/id= /\*Optionally, with view=CASE: string case identifier variable

/missing= /\*What to do with user-missing data: EXCLUDE (default) – exclude the cases listwise

/\*(like system-missing); INCLUDE - take user-missing as valid data

/normalize= NONE /\*Normalize input vectors: no (NONE, default), to sum 1 (SUM),

/\*to sum of squares 1 (SS), z-standardization (Z), to range of 1 (RANGE),

/\*to range 0-1 (RESCALE), sofmax (EXP), centered logratio (CLR),

/\*z-standardization transverse (ZDIM)

/measure= CHORD /\*Proximity measure: EUCLID, BLOCK, MINK p, CHEB, CANBERRA, CLARK, BRAYCURT,

/\*SOERGEL, KULCZ1, HEDGES, KULCZ2, PINKH i, ROBERTS, INTERSECT, HELLINGER, COS,

/\*CHORD, ARCCOS, SIN, SIMRATIO, IDENTC, HMEAN, GMEAN, BHATTACH, MORHORN,

/\*ELLENBERG, GLEASON, PANDEYA, MAHAL, MAHAL2, CHISQPR, PNDIV, SPNDIV,

/\*KLDIV, SKLDIV, KDIV, SKDIV, JDIFF, TANEJA, SIZE, SCATTER, SHAPE, ZSHAPE

/\*CATRP, MCRPA, ICC, BUTLER, BUTLER2

/matrix= *'d:\exercise\covmx.sav'* /\*Optional, for MEASURE= MAHAL, MAHAL2, BUTLER, BUTLER2:

/\*Use this matrix (filename)

/noroot= /\*Don’t take the root from measures that are a root: YES or NO (default)

/abs= /\*Abolish negative sign in computed proximities: YES or NO (default)

/reverse= /\*Convert proximities: similarities in dissimilarities, dissimilarities in

/\*similarities: don’t do (NONE, default), NEGSHIFT, ONEMINUS, LAWCOS, RECIP k (see).

Minimal specification VARS, MEASURE.

The macro computes some proximity measures (similarities, dissimilarities/distances) between vectors of data, pairwise, and outputs square matrix as new unnamed dataset. Items – vectors between which a proximity is computed – may be cases (rows) or variables (columns) of the input dataset.

SPSS Statistics command PROXIMITIES computes some basic proximity measures for quantitative data (such as Euclidean distance, Minkowski distance, Manhattan, correlation coefficient, cosine, chi-square, etc.). The present macro offers much richer choice of proximity measures.

Input data must be quantitative – scale or count/proportion; binary data (1 vs 0) are also permitted as a particular case. There must be no variables named *CASENO#$* and *STRID#$* in the input dataset*.*

If distances (dissimilarities) were computed, the output matrix has 0 on the diagonal and nonnegative off-diagonal values. If similarities were computed, there will be positive values (not always just 1) on the diagonal, while off-diagonal values may or may not have negative sign depending on the type of proximity.

Column *PROX* in the matrix has value label starting by word “DISSIMILARITY” if the proximity measure is dissimilarity, and word “SIMILARITY” if the proximity is similarity. This is important, because SPSS Statistics procedures taking a matrix in analysis (for example, cluster), look on that first word. So if you want to convert dissimilarities to similarities or similarities to dissimilarities – don’t forget to change that word in the label of column *PPOX*. The macro has s/c REVERSE converting dissimilarities and similarities one into another by some popular methods, however, there exist many other methods of such conversion, so what REVERSE offers may not suit you; you must decide what reverse transform is optimal for your data and your proximity measure.

The procedure computes a measure for all pairs of vectors simultaneously. Therefore, if some of the pairs do not permit the given measure to be calculated, the whole matrix won’t be computed. Error message with the reason why it was not possible to compute – will be issued.

***Subcommands***

**VARS**

Specify by list and/or via “to” input quantitative numeric variables.

**VIEW**

Specify what constitute vectors, items between which proximity to compute. Specify VIEW=CASE if the proximities need to be between cases, data rows (also default), or VIEW=VARIABLE if the proximities need to be between variables, data columns. In the second instance, names of columns/rows of the output matrix will be the names of variables VARS. In the first instance, names of columns/rows of the matrix will be successive VAR1, VAR2, VAR3…, and there will be present column *CASENO*\_ - the case number in the initial dataset.

**MISSING**

The macro deletes missings listwise: if a case has missing value at least in one variable VARS or in the identifier, it is not taken in the analysis. Note that at input there are deleted cases, not variables, irrespective of how s/c VIEW is specified.

Subcommand MISSING allows to take in the analysis cases with user-missing values, i.e. to treat them as valid: MISSING=INCLUDE. By default and by MISSING=EXCLUDE they are treated as missing and the cases are deleted, like system-missing.

**ID**

This subcommand is allowed only when VIEW=CASE. You may specify string variable (up to 8 bytes wide) to serve as case identifier in the output matrix of coefficients. By default/unspecification of the subcommand the macro will create string identifier named *ID*, which values are Latin “C” plus the case number coinciding with *CASENO*\_.

**NORMALIZE**

Optional subcommand, it normalizes values in vectors before computing proximities.

NONE - (also default/unspecification) don’t do, use values “as is”. Note, however, that some measures (see s/c MEASURE) perform themselves the linear normalizing to sum (SUM) as part of their formula.

SUM - linear normalizing to sum: in each vector values are divided by the sum in the vector; the sum in the vector will become 1. This option requires nonnegative data. If a vector was zero the macro leaves it such.

SS - normalizing to sum of squares: in each vector values are divided by the square root of the sum of squares in the vector; sum of squares in the vector will become 1. If a vector was zero the macro leaves it such.

Z - z-standardization: in each vector the mean will be 0 and the st. deviation 1. If a vector was a constant it will become zero vector.

RANGE - taking to range of 1: in each vector values are divided by the range in the vector. If a vector was a constant the macro leaves it as is.

RESCALE - linear rescaling to range [0,1]: in each vector its minimal value is subtracted from values, then the values are divided by the range. If a vector was a constant the macro makes it constant 0.5.

EXP - exponential normalization, or softmax transform. In each vector values will lie in the range 0–1 and their sum will be 1.

CLR - centered logratio transformation. First in each vector normalizing to sum is done (SUM). In the obtained vector, values are divided by the geometric mean in it, and logarithm is taken; sum in the vector will become 0. This option requires positive input data.

ZDIM - z- standardization “transverse”, i.e. within dimensions, not vectors: in each dimension the mean will be 0 and the st. deviation 1. If a dimension was zero the macro leaves it such. Unlike Z, the normalizing standard deviation will be computed here on “df=n”, not “df=n-1”. (With VIEW=CASE vectors are cases, and dimensions are variables of the data, while with VIEW=VARIABLE it is opposite.)

**MEASURE**

Select one proximity measure (similarity or dissimilarity) from the list. Notations *x* and *y* are the corresponding values of the two vectors between which the proximity is being computed. I.e. it is *xi* and *yi*, however indices “*i*” are omitted in the formulas for the sake of shorthand. is , where *m* is the number of “dimensions”, that is, the number of elements in vector. By word “dimension” there then implied are data columns – when vectors are rows (VIEW=CASE), or data rows – when vectors are columns (VIEW=VARIABLE). means, consequently, ; means .

Designations for linearly normalized to unit sum nonnegative vectors (i.e., compositional data): and .

EUCLID - **Euclidean distance**,

Note: euclidean distance after NORMALIZE=CLR is known as **Aitchison distance**. After NORMALIZE=SS is known as **chord distance**.

BLOCK - **Manhattan** aka **city-block distance**,

Manhattan distance after NORMALIZE=SUM, and divided by 2, is known as **Whittaker distance**.

MINK *p* - **Minkowski distance**,

where arbitrary degree *p* – number in range (0,+∞]. *p*=1 gives **Manhattan** distance, *p*=2 gives Euclidean distance.

CHEB - **Chebyshev distance** aka **chessboard distance**, aka **supremum distance**,

CANBERRA - **Canberra distance**,

(if denominator is 0 the entire term is assumed 0).

CLARK - **Clark distance** (aka **Clark divergence coefficient**). This is L2 version of Canberra distance (which, in turn, is L1 version),

(if denominator is 0 the entire term is assumed 0).

BRAYCURT - **Bray–Curtice distance**, for nonnegative data,

Similarity 1-*dxy* has many synonymic names: **Czekanowski similarity**, **Bray–Curtice similarity**; applied to binary data – **Sorensen** or **Dice similarity**; in analysis of 2x2 confusion tables – **F1 index**; in analysis of proportions – **Renkonen index**. Half the 1-*dxy* is called **Motyka similarity**. *dxy* itself in case of binary data bears name **Lance–Williams dissimilarity**.

ELLENBERG - **Ellenberg similarity**, for nonnegative data,

where function equals 1 when *xy* is not 0, otherwise it equals 0.

GLEASON - **Gleason similarity**, for nonnegative data. Same formula as Ellenberg, but without “” in denominator.

PANDEYA - **Pandeya similarity**, for nonnegative data,

where function as in Ellenberg above.

KULCZ1 - **Kulczynski dissimilarity 1**, for nonnegative data,

1-*dxy* is **Kulczynski similarity 1**.

SOERGEL - **Soergel dissimilarity** (aka **Tanimoto distance** aka **Levandowsky distance** aka **Marczewski–Steinhaus coefficient**), for nonnegative data,

1-*dxy* is called **Ruzicka similarity** and also quantitative data **Jaccard similarity** (another pretender is *SIMRATIO*).

HEDGES - **Hedges wave dissimilarity**, for nonnegative data,

KULCZ2 - **Kulczynski dissimilarity 2**, for nonnegative data,

1-*dxy* is **Kulczynski similarity 2**.

PINKH *i* - **Pinkham–Pearson similarity**, aka **fuzzy Jaccard similarity**, for nonnegative data,

(if denominator is 0 the entire term is assumed *i*; *i* is the digit either 0 or 1 following the keyword).

ROBERTS - **Roberts similarity**, for nonnegative data,

INTERSECT - **intersection similarity**, for nonnegative data,

1-*dxy* is **intersection distance**.

MAHAL - **Mahalanobis distance** (aka **generalized distance**) using covariance matrix,

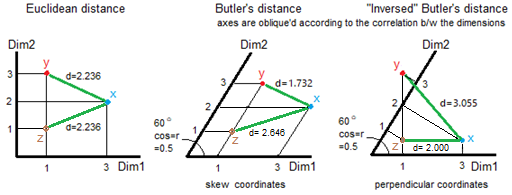
where *S* is the covariance matrix between the dimensions. Requires nonsingular *S*. Mahalanobis distance is euclidean distance obtained upon “removal” from data information about correlation and about variance (scale) of dimensions.

To input your own matrix *S* rather than compute it from the data – see s/c MATRIX.

MAHAL2 - **Mahalanobis distance** using MSCP matrix. Same formula as *MAHAL*, but *S* is the matrix of mean crossproducts between the dimensions, i.e. it is like covariance matrix only without centering of data at its computation.

BUTLER - **Butler distance** using correlation matrix,

where *R* is the correlation matrix between the dimensions. Butler’s distance is euclidean distance in oblique axial frame: angles between the axes are set by *R*. Input data (*x* and *y*) are taken for contravariant (skew) coordinates.



To take input data for covariant (perpendicular) coordinates, add keyword INV after word BUTLER. Then will be used in place of in the formula ( must be nonsingular then); this “inverted” Butler distance is the Mahalanobis without scaling.

To input your own matrix *R* rather than compute it from the data – see s/c MATRIX.

BUTLER2 - **Butler distance** using cosine matrix. Same formula as *BUTLER*, but *R* is the matrix of cosine similarities between the dimensions, it is like correlation matrix only without centering of data at its computation. Add keyword INV to treat your data as covariant coordinates.

COS - **cosine similarity** (aka **Tucker coefficient of congruence** aka **coefficient of proportionality** aka **Orchini similarity**),

1-*sxy* is **cosine distance**. After NORMALIZE=Z cosine is Pearson correlation coefficient.

CHORD - **chord distance** (aka **Orloci distance**),

This is euclidean distance between NORMALIZE=SS vectors. In other words, this is cosine similarity converted to euclidean distance by cosine theorem.

ARCCOS - cosine converted to **angular distance** (aka **geodesic metric**) – angle in radians,

SIN - **sine dissimilarity**,

IDENTC - **Zegers–ten Berge identity coefficient similarity**, aka **quantitative Dice similarity**,

SIMRATIO - **similarity ratio** (aka **Kohonen similarity** aka **Kumar–Hassebrook** **similarity**),

This measure can be seen **quantitative data Jaccard similarity** (another pretender is 1-*SOERGEL*).

MORHORN - **Morisita–Horn similarity**, for nonnegative data,

HMEAN - **harmonic mean similarity**, for nonnegative data,

Used mainly for vectors of proportions or probabilities with unit sum in vector. Then *sxx*=1.

GMEAN - **geometric mean similarity**, for nonnegative data,

Used mainly for vectors of proportions or probabilities with unit sum in vector. Then *sxx*=1 and the measure is known as **fidelity** or **Bhattacharyya coefficient** or **Hellinger affinity**. To obtain **fidelity**, use NORMALIZE=SUM.

BHATTACH - **Bhattacharyya distance**, for nonnegative data,

This is negative logarithm of **fidelity**.

HELLINGER - **Hellinger distance**, for nonnegative data,

, i.e., without multiplier 2 under the root, is called **Matusita** or **Jeffries–Matusita distance[[1]](#footnote-1)**.

CHISQPR - **chi-square distance between two probability distributions**, for nonnegative data,

This is square root of the chi-square statistic of cross-table 2 x *m* (vectors x categories) where sums within vectors were taken to unit[[2]](#footnote-2). is known as **probabilistic symmetric chi-square**.

PNDIV - **Pearson/Neyman chi-square of disagreement of probabilities** (aka Pearson/Neyman divergence), for positive data,

If is the observed probability distribution and is the expected (theoretical) probability distribution then is the square root of the chi-square statistic of comparison of the observed against the expected probabilities. The macro returns asymmetric matrix in which element (r,c) corresponds to instance when the rth vector is in the denominator.

SPNDIV - **symmetric Pearson–Neyman chi-square of disagreement of probabilities**, for positive data. This is . The macro returns symmetric matrix.

KLDIV - **Kullback–Leibler divergence** (aka **KL-distance** aka **relative entropy** aka **information deviation** aka **information gain**), for nonnegative data,

(if or the entire term is assumed 0). The macro returns asymmetric matrix in which element (r,c) corresponds to instance when the cth vector is , rth vector is .

SKLDIV - **symmetric Kullback–Leibler divergence** (aka **Jeffreys divergence** aka **J-distance**), for nonnegative data. This is , where *d* is the Kullback–Leibler above. The macro returns symmetric matrix.

KDIV - **K-divergence**, for nonnegative data,

(if or the entire term is assumed 0). The macro returns asymmetric matrix in which element (r,c) corresponds to instance when the cth vector is , rth vector is .

SKDIV - **symmetric** **K-divergence** (aka **Topsoe distance**), for nonnegative data. This is , where *d* is the K-divergence above. The macro returns symmetric matrix. Half of Topsoe distance is known as **Jensen–Shannon divergence**.

JDIFF - **Jensen difference** (aka **information radius**), for nonnegative data,

(if or the entire term is assumed 0).

TANEJA - **Taneja distance**, for nonnegative data,

(if or the entire term is assumed 0).

SIZE - **Penrose size difference**,

It is (divided by ) summed difference between vectors and expresses the difference of *levels* (elevations) of the profiles. Symmetric elements in the matrix will differ in sign. Element (r,c) corresponds to instance when the cth vector is *x*, rth vector is *y*. S/c ABS=YES will remove sign from elements.

SCATTER - **scatter difference**,

where *cx* and *cy* are centered vectors *x* and *y*. It is inequality of the magnitude of *scatter* in the profiles. Symmetric elements in the matrix will differ in sign. Element (r,c) corresponds to instance when the cth vector is *x*, rth vector is *y*. S/c ABS=YES will remove sign from elements.

SHAPE - **Penrose shape distance**,

This quantity is equal to the root square deviation of differences *x-y* from 0, i.e. to the euclidean distance between centered vectors (*cx* and *cy*), and it is a measure of nonequality of *shape* of profiles *x* and *y*.

ZSHAPE - **standardized shape distance**,

This quantity is equal to the euclidean distance between z-standardized vectors (*zx* and *zy*), and it is a measure of nonequality of *shape* of profiles *x* and *y*. It linearly is equivalent to the uncorrelatedness, i.e. to .

CATRP - **Cattell pattern similarity**, one of measures of profile likeness,

where is euclidean distance between the two vectors; *q* is the median in chi-square distribution with df=*m*, computed by the macro by approximate formula . Similarity normally expects that values of input vectors are z-standardized within *dimensions* (using statistics of the given sample or external normatives). Use NORMALIZE=ZDIM to standardize within dimensions by the sample’s statistics.

MCRPA - **McCrae profile agreement**, one of measures of profile likeness,

where is euclidean distance between the two vectors. Similarity normally expects that values of input vectors are z-standardized within *dimensions* (using statistics of the given sample or external normatives). Use NORMALIZE=ZDIM to standardize within dimensions by the sample’s statistics.

ICC - **intraclass correlation coefficient**, one of measures of profile likeness,

where is the total mean of the combined array of the two vectors.

In case of *binary data*, Similarity ratio = Jaccard sim = Ruzicka sim (= 1 - Soergel dis) = Ellenberg sim.

In case of *binary data*, Pinkham-Pearson (1) = Rand sim; Pinkham-Pearson (0) = Russel-Rao sim.

In case of *binary data*, Identity coef = Dice sim = Czekanowski sim (= 1 - Bray-Curtis \ Lance-Williams dis) = Gleason sim.

In case of *binary data*, Cosine sim = Ochiai sim = Fidelity.

One convenient, practical **classification** of the measures:

* **Measures for data with any values**. Euclidean distance, Manhattan distance, Minkowski distance, Chebyshev distance, Canberra distance, Clark distance, Cosine similarity / Pearson correlation, Chord distance, Arc cosine distance, Sine dissimilarity, Identity coefficient similarity, Similarity ratio.
* **Measures for data with nonnegative values**. Bray–Curtice distance, Ellenberg similarity, Gleason similarity, Pandeya similarity, Kulczynski dissimilarity 1, Kulczynski dissimilarity 2, Soergel dissimilarity, Hedges wave dissimilarity, Pinkham–Pearson similarity (with parameter 0 or 1), Roberts similarity, Intersection similarity, Morisita–Horn similarity, Harmonic mean similarity, Geometric mean similarity.
* **Measures often used for probability vectors** or other “compositional data” such as counts, with nonnegative values summing up to 1 or to some Total in a vector (the macro first normalizes so that they sum to 1). Bhattacharyya distance, Hellinger distance, Chi-square distance for probabilities, Pearson/Neyman chi-square divergence (asymmetric and symmetric), Kullback–Leibler asymmetric divergence (Information gain), Kullback–Leibler symmetric divergence (Jeffreys divergence), K-divergence (asymmetric), K-divergence (symmetric, Topsoe distance), Jensen difference (Information radius), Taneja distance. Harmonic mean similarity and Geometric mean similarity are also used for probability vectors in conjunction with NORMALIZE=SUM. Aitchison distance is Euclidean distance computed on centered-logratio-transformed vectors.
* **Measures often used to assess profile likeness**, to compare “profiles”. Size difference, Scatter difference, Shape distance, Standardized shape distance, Cattell pattern similarity, McCrae profile agreement, Intraclass correlation coefficient. Butler’s distance is used for profile comparison, too.
* **Measures which consider correlations**. These are distances which when computed between, say, cases, simultaneously take account of correlations between the variables. Mahalanobis distance, Butler’s distance. (Butler amplifies distances co-directed with the correlation and condenses distances counter-directed with it, while Mahalanobis distorts it the opposite, and also scales.)

Sources:

* Deza, M.M, Deza, E. Encyclopedia of distances. Springer, 2009.
* Cha, S.-H. Comprehensive survey on distance/similarity measures between probability density functions // *International Journal of Mathematical Models and Methods in Applied Sciences*, 2007, 4(1), 300-307.
* Podani, J. Introduction to the exploration of multivariate biological data. Backhuys Publishers, 2000. Chapter 3.
* McCrae, R.R. A Note on Some Measures of Profile Agreement // *Journal of Personality Assessment*, 2008, 90(2), 105-109.
* Rifkin, S. Comparing patterns: elevation, scatter, and shape / Paper for Executive Leadership Program course in the Graduate School of Education, George Washington University, 2006.
* Butler, J.K. A vector model for describing and comparing profiles // Educational and Psychological Measurement, 1983, 43(3), 747-758.

**MATRIX**

This optional subcommand acts only with MEASURE= *MAHAL*, *MAHAL2*, *BUTLER*, or *BUTLER2*. You can enter matrix *S* or *R* (see formulas in s/c MEASURE) for computation of the indicated distances. The entered matrix, and not the matrix computed from the data, will be used then. Specify an external .SAV file in quotes or apostrophes. The file must present itself as a square symmetric matrix with number of rows and columns equal to the number of dimensions *m* of the data. Names of the *m* variables of the file can be any. When you input your matrix rather than let it be computed from the data, there is no difference in results between *MAHAL* and *MAHAL2*, between *BUTLER* and *BUTLER2*.

**NOROOT**

NOROOT=YES returns a measure which is a root, without taking the root from it. For example, MEASURE=*EUCLID* under NOROOT=YES will returm squared Euclidean distance, not Euclidean distance, because the last action in the formula of Euclidean distance – taking of the root – will be withdrawn. This subcommand concerns the following measures: *EUCLID*, *CLARK*, *MAHAL*, *MAHAL2*, *BUTLER*, *BUTLER2*, *CHORD*, *SIN*, *CHISQPR*, *PNDIV*, *SPNDIV*, *SHAPE*, *ZSHAPE*, *MINK*. Other measures ignore it.

**ABS**

Optional subcommand abolishing negative sign in computed proximities (if the given proximity measure can be negative): ABS=YES.

**REVERSE**

This subcommand reverses computed proximities: similarities into dissimilarities, dissimilarities into similarities. Choose method:

NONE - (also default/unspecification) don’t reverse.

NEGSHIFT - change sign and then shift towards positive side. This is a universal linear method, it is suitable for all proximities. Similarities are transformed into dissimilarities by the formula , where is the highest off-diagonal similarity in the matrix, and on the diagonal of the result dissimilarity matrix 0 is set. Dissimilarities are transformed into similarities by the formula , where is the greatest dissimilarity in the matrix.

ONEMINUS - subtract from one. This method transforms similarities into dissimilarities by the formula and is suitable for similarities with values in the range [0, 1] or [-1, 1]. It transforms dissimilarities into similarities by the formula and is suitable for dissimilarities with values in the range [0, 1].

LAWCOS - this method is based on the law of cosines. It transforms similarities into dissimilarities by the formula and is suitable for similarities with values in the range [0, 1] or [-1, 1]. It transforms dissimilarities into similarities by the formula and is suitable for dissimilarities with values in the range [0, 1].

RECIP *k* - through inversion. This method transforms similarities into dissimilarities by the formula and is suitable for similarities with values in the range (0, 1]. It transforms dissimilarities into similarities by the formula and is suitable for any dissimilarities. *k* is a positive number. By default of the number *k* it is taken for 1.

When ABS=YES then REVERSE is applied after ABS.

***Special regimes***

The macro ignores weighting of the dataset (however, it doesn’t take in the procedure cases with missing and nonpositive weights) and is not suited for splitting the dataset. The macro obeys case selection/filtering (commands FILTER, USE, SELECT IF, N OF CASES) and temporary (under TEMPORARY) transformations. If you use selection by commands SELECT IF or N OF CASES not under TEMPORARY, you’ve removed a portion of cases from the original dataset and the values of column CASENO\_ by output matrix are then the case numbers in that remaining dataset. In all other screening regimes the values of CASENO\_ are the case numbers in the original dataset observed there prior your screening of cases.

# MACRO !KO\_PROXBIN: VARIOUS PROXIMITY MEASURES FOR BINARY DATA

Version 1, Oct 2020. Tested on SPSS Statistics 20, 22, 25.

!KO\_proxbin vars= *v1 to v15* /\*List of binary variables (may use “to”)

/view= CASE /\*Compute proximities between cases (CASE, default) or

/\*between variables (VARIABLE)

/id= /\*Optionally, with view=CASE: string case identifier variable

/measure= SM /\*Proximity measure: COUNT\_A, COUNT\_B, COUNT\_C, COUNT\_D,

/\*ECOUNT\_A, ECOUNT\_B, ECOUNT\_C, ECOUNT\_D,

/\*RR, SM, HAMANN, JACCARD, DICE, LW, RT, SS1, SS2, K1, SS3,

/\*K2, MCCON, OCHIAI, SS4, SS5, GOWER, CHISQ, CHISQY, CC, PHI, PRPHI, DISPER,

/\*SEUCLID, EUCLID, SIZE, SHAPE, PATTERN, VARIANCE, MN, TETRACH, LAMBDA,

/\*D, Y, Q, DIGBY, GK1, SIMPSON, BRAUN, FAI1, FAI2, BUB1, BUB2, MOUNTF, QDIFF,

/\*FCOS, CHORD, HELLINGER, MICHAEL, HD, FORBES, ALROY, PMI, DENNIS,

/\*TARWID, KAPPA, SPI, LOEV, COLE, MAXWELL, FLEISS, EYRAUD, GILBERT

/adjust= /\*For similarities: rescale computed measure values:

/\*don’t do (NONE, default), INDBOUND, INDUPPER, MARGPOLE,

/\*INDMARGPOLE, INDMARGMAX (see)

/abs= /\*Abolish negative sign in computed proximities: YES or NO (default)

/reverse= /\*Convert proximities: similarities in dissimilarities, dissimilarities in

/\*similarities: don’t do (NONE, default), NEGSHIFT, ONEMINUS, LAWCOS, RECIP k (see).

Minimal specification VARS, MEASURE.

The macro computes some proximity measures (similarities – including matching coefficients, dissimilarities/distances) between vectors of binary data, pairwise, and outputs square matrix as new unnamed dataset. Items – vectors between which a proximity is computed – may be cases (rows) or variables (columns) of the input dataset.

There must be no variables named *CASENO#$* and *STRID#$* in the input dataset*.*

If distances (dissimilarities) were computed, the output matrix has 0 on the diagonal and nonnegative off-diagonal values. If similarities were computed, there will be positive values[[3]](#footnote-3) (not always just 1) on the diagonal, while off-diagonal values may or may not have negative sign depending on the type of proximity.

Column *PROX* in the matrix has value label starting by word “DISSIMILARITY” if the proximity measure is dissimilarity, and word “SIMILARITY” if the proximity is similarity. This is important, because SPSS Statistics procedures taking a matrix in analysis (for example, cluster), look on that first word. So if you want to convert dissimilarities to similarities or similarities to dissimilarities – don’t forget to change that word in the label of column *PPOX*. The macro has s/c REVERSE converting dissimilarities and similarities one into another by some popular methods, however, there exist many other methods of such conversion, so what REVERSE offers may not suit you; you must decide what reverse transform is optimal for your data and your proximity measure.

The procedure computes a measure for all pairs of vectors simultaneously. Therefore, if some of the pairs do not permit the given measure to be calculated, the whole matrix won’t be computed. Error message with the reason why it was not possible to compute – will be issued.

***Subcommands***

**VARS**

Specify by list and/or via “to” input binary numeric variables having only two valid values: 1 and 0. If your data are not dichotomous or bear not binary coding, recode them first into binary ones by RECODE command, what you could do also temporarily (under TEMPORARY). If your binary data represent the binary/boolean status of an attribute, than value 1 must correspond to the status “present” or “true”, while 0 to the status “absent” or “false”.

The macro deletes any missings listwise: if a case has missing value at least in one variable VARS or in the identifier, it is not taken in the analysis. Note that at input there are deleted cases, not variables, irrespective of how s/c VIEW is specified.

**VIEW**

Specify what constitute vectors, items between which proximity to compute. Specify VIEW=CASE if the proximities need to be between cases, data rows (also default), or VIEW=VARIABLE if the proximities need to be between variables, data columns. In the second instance, names of columns/rows of the output matrix will be the names of variables VARS. In the first instance, names of columns/rows of the matrix will be successive VAR1, VAR2, VAR3…, and there will be present column *CASENO*\_ - the case number in the initial dataset.

**ID**

This subcommand is allowed only when VIEW=CASE. You may specify string variable (up to 8 bytes wide) to serve as case identifier in the output matrix of coefficients. By default/unspecification of the subcommand the macro will create string identifier named *ID*, which values are Latin “C” plus the case number coinciding with *CASENO*\_.

**MEASURE**

Select one proximity measure (similarity or dissimilarity) from the list. Let two compared vectors be X and Y, both of length *m* – that is the number of “dimensions” (if X and Y are data rows, then dimensions are data columns, and if X and Y are data columns, then dimensions are data rows). Then the notations in the formulas are:

*a* number of dimensions where X=1 and Y=1

*b* number of dimensions where X=1 and Y=0

*c* number of dimensions where X=0 and Y=1

*d* number of dimensions where X=0 and Y=0

*a+b+c+d=m*

I.e. we have the frequency table:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Vector Y | |
|  |  | 1 | 0 |
| Vector X | 1 | *a* | *b* |
| 0 | *c* | *d* |

This table of four counts is displayed in the Output Viewer if the number of vectors in the analysis equals just two.

Measures below where the range is not expressed with the formula vary in the range [0,1]. On the diagonal of the matrix of measures *RR*, *GOWER*, *CHISQY*, *DISPER*, *D*, *FAI2*, *FAI1*, *FCOS*, *MICHAEL*, *FORBES*, *PMI*, *TARWID*, *DENNIS*, *COLE*, *EYRAUD* there stands not a constant, because in these similarity measures the value of *sii* depends on the shape of distribution in vector *i*. Moreover, in some of these measures there can encounter 0 on the diagonal.

*Note* on Sokal–Sneath measures 1, 2, 3, 4, 5 (i.e. *SS1, SS2, SS3, SS4, SS5*): in different literature sources there exists confusion which number which formula is possessed by, while the formulas are the same in all the sources. In this macro, formula-number bind follows that in SPSS Statistics documentation (commands PROXIMITIES, CLUSTER). A user is recommended to look at the formula, not at the number.

RR - **Russel–Rao similarity** aka **simple joint probability**,

SM - **Rand similarity** aka **simple matching** aka **Sokal–Michener similarity**,

HAMANN - **Hamann similarity** is *SM* recalculated from range [0, 1] into [-1, 1],

JACCARD - **Jaccard similarity** aka **Tanimoto similarity** aka **community coefficient** aka **similarity ratio**,

1-*JACCARD* is called **Soergel distance**.

DICE - **Dice similarity** aka **Czekanowski similarity** aka **Sorenson similarity** aka **Gleason similarity** aka **identity coefficient** aka **coincidence index**,

It is the harmonic mean of two conditional probabilities: and (in classification tasks known as “Recall” and “Precision”).

LW - **Lance­–Williams** dissimilarity,

It is the **Bray–Curtis distance** in case of binary data.

RT - **Rogers–Tanimoto similarity**,

SS1 - **Sokal–Sneath 1 similarity** aka **Gower–Legendre similarity**,

SS2 - **Sokal–Sneath 2 similarity**,

K1 - **Kulczynski 1 similarity**. Ranges in [0, *m*-1]. Values *sii* are incomputable and arbitrary 999 is put on the diagonal.

SS3 - **Sokal–Sneath 3 similarity**. Ranges in [0, *m*-1]. Values *sii* are incomputable and arbitrary 999 is put on the diagonal.

K2 - **Kulczynski 2 similarity**,

It is the arithmetic mean of two conditional probabilities: and (in classification tasks known as “Recall” and “Precision”).

MCCON - **McConnaughey similarity**, is *K2* recalculated from range [0, 1] into [-1, 1],

OCHIAI - **Ochiai similarity** aka **cosine similarity** aka **Otsuka similarity**,

It is the geometric mean of two conditional probabilities: and (in classification tasks known as “Recall” and “Precision”). *OCHIAI*2 is known as **Sorgenfrei similarity** aka **correlation ratio**.

SS4 - **Sokal–Sneath 4 similarity** aka **Anderberg 2 similarity**,

It is averaging of two similarities Kulczynski 2 (*KK2*): one corresponding to data coding “1=present, 0=absent” (as usual), and the other corresponding to the reverse coding.

SS5 - **Sokal–Sneath 5 similarity** aka **Anderberg 1 similarity** aka **Ochiai 2 similarity**,

It is the product of two cosines (*OCHIAI*): one corresponding to data coding “1=present, 0=absent” (as usual), and the other corresponding to the reverse coding.

GOWER - **binary Gower similarity**. Ranges in [0, *m*/(*m*-1)].

CHISQ - **Pearson chi-square similarity**, the chi-square statistic of 2×2 table. Ranges in [0, *m*].

CHISQY - **Yates corrected Pearson chi-square** similarity, the chi-square statistic of 2×2 table. Ranges in [0, *m*).

Logarithm of *CHISQY* is known as **Stiles similarity**.

CC - **Contingency coefficient** similarity, ranges (for 2×2 table) in [0, √0.5].

PHI - **Phi correlation** similarity is the **Pearson correlation** coefficient in situation of binary data Ranges in [-1, 1],

where is , if , else is . *PHI*2 is known also as **Doolittle similarity**. *PHI* in case of binary data coincides with **Cramer’s** **V**, **Spearman rho**, **Kendall tau-b**.

PRPHI - **pole-rescaled Phi correlation** aka **Cole’s C7 similarity**, ranges in [-1, 1],

where is the maximal by absolute value positive (if *PHI* is positive) or negative (if *PHI* is negative) phi-correlation attainable in the given 2×2 table with fixed marginal frequencies; is the magnitude of chi-square corresponding to , and is as in *PHI*.

and , under

and , under (see Warrens, 2008).

*PRPHI* is the same thing as *PHI* with subcommand ADJUST=MARGPOLE.

DISPER - **dispersion similarity**, ranges in [-0.25, 0.25].

It is the (biased) **covariance** in case of binary data.

SEUCLID - **squared Euclidean distance** aka **Hamming distance**, ranges in [0, ∞],

For binary data it coincides with **Manhattan distance** and **Canberra distance**. 2*SEUCLID* is called **Mirkin distance**.

EUCLID - **Euclidean distance**, ranges in [0, ∞],

SIZE - **size difference** dissimilarity aka **Baulieu dissimilarity**,

It is squared Penroze size difference, one of measures of profile comparison.

SHAPE - **shape difference** dissimilarity,

It is squared Penroze shape difference, one of measures of profile comparison.

PATTERN - **pattern difference** dissimilarity,

2√*PATTERN* is called **Sneath dissimilarity**.

VARIANCE - **variance** **dissimilarity**,

MN - **McNemar distance** is the square root of McNemar test statistic. Ranges in [0, ∞].

TETRACH - **tetrachoric correlation**, Pearsonian approximation, ranges in [-1, 1],

where is , if , else is . “Min” aims to make the coefficient symmetric over recoding data 1 into 0 and 0 into 1, – what is expected from a correlation coefficient.

LAMBDA - **Goodman–Kruskal lambda** similarity,

D - **Anderberg’s D similarity**, ranges in [0, 0.5].

where and as in *LAMBDA.*

Q - **Yule’s Q** similarity aka **Yule’s coefficient of association**, ranges in [-1, 1],

where *OR* is the odds ratio .

It is **Goodman–Kruskal gamma** in case of binary data.

Y - **Yule’s Y** similarity aka **Yule’s coefficient of colligation**, ranges in [-1, 1],

where *OR* as in *Q*.

DIGBY - **Digby similarity** is the “compromise between” *Q* and *Y*, ranges in [-1, 1],

This coefficient can serve an approximation to tetrachoric correlation, alternative to Pearsonian approximation *TETRACH*.

GK1 - **Goodman–Kruskal 1 similarity**, ranges in [-1, 1],

SIMPSON - **Simpson similarity** aka **overlap similarity**,

BRAUN - **Braun–Blanquet similarity** aka **Savage similarity**,

(*BRAUN + SIMPSON*)/2 *= K2*.

FAI2 - **Faith 2 similarity**,

FAI1 - **Faith 1 similarity** is *FAI2* recalculated from range [0, 1] into [-1, 1],

BUB2 - **Baroni-Urbani & Buser 2 similarity**,

It is a modification of *SM* (Rand), where *d* is replaced with . As approaches zero the measure approaches to *JACCARD*.

BUB1 - **Baroni-Urbani & Buser 1 similarity** is *BUB2* recalculated from range [0, 1] into [-1, 1],

It is a modification of *HAMANN*, where *d* is replaced with .

MOUNTF - **Mountford similarity**. Values *sii* are incomputable and arbitrary 999 is put on the diagonal.

This measure is not computed when *b* or *c* is 0.

QDIFF - **Q0 difference** dissimilarity, ranges in [0, ∞],

FCOS - **Fager’s corrected cosine**. It is cosine (*OCHIAI*) diminished by the subtracted quantity:

If value is less than 0 it is zeroed.

CHORD - **chord distance**, ranges in [0, √2],

It is cosine reversed into corresponding Euclidean distance.

HELLINGER - **Hellinger distance**, ranges in [0, 2],

MICHAEL - **Michael similarity**, ranges in [-1, 1],

HD - **Hawkins–Dotson similarity**,

It is averaging of two Jaccard similarities (*JACCARD*): one corresponding to data coding “1=present, 0=absent” (as usual), and the other corresponding to the reverse coding.

FORBES - **Forbes similarity**, aka Mozley similarity [wrong attribution]), ranges in [0, *m*],

where .

PMI - **point** aka **binary mutual information** aka **Gilbert–Wells similarity**, ranges in [-∞, ln *m*],

where and *.*

ALROY - **Alroy similarity** aka **modified Forbes similarity**,

where .

TARWID - **Tarwid similarity**, ranges in [-1, (*m*-1)/(*m*+1)],

where as in *FORBES*.

DENNIS - **Dennis similarity**, ranges in [-*m*/(2√*m*), (*m*-1)/√*m*],

KAPPA - **Cohen’s kappa** similarity, for binary data ranges in [-1, 1],

This measure is also known as **Adjusted Rand** similarity and is the same thing as *SM* with subcommand ADJUST= INDMARGMAX.

SPI - **Scott’s Pi** similarity, for binary data ranges in [-1, 1],

LOEV - **Loevinger similarity** aka **Benini similarity**, ranges in [-(*m*-1), 1],

COLE - **Cole similarity**, ranges in [-1, *m*-1].

MAXWELL - **Maxwell–Pilliner similarity**, ranges in [-1, 1],

FLEISS - **Fleiss similarity** [not to confuse with Fleiss kappa], ranges in [-((*m*-1)/2+1/(*m*-1)), 1],

EYRAUD - **Eyraud similarity**, ranges in [-*m*2/(*m*-1),*m*2/(*m*-1)].

where as in *FORBES* and .

GILBERT - **Gilbert similarity**, ranges in [-1, 1].

where as in *FORBES*.

MEASURE subcommand can also return the *observed* counts *a*, *b*, *c* or *d*, or *expected* (under hypothesis of independence) counts *E(a)*, *E(b)*, *E(c)* or *E(d)*. An expected count is the product of two respective marginal counts, divided by the total count; for example, .

COUNT\_A - matrix of counts *a*.

COUNT\_B - matrix of counts *b*. It is asymmetric matrix.

COUNT\_C - matrix of counts *c*. It is asymmetric matrix coinciding with *COUNT\_B* transposed.

COUNT\_D - matrix of counts *d*.

ECOUNT\_A - matrix of expected counts *E(a)*.

ECOUNT\_B - matrix of expected counts *E(b)*. It is asymmetric matrix.

ECOUNT\_C - matrix of expected counts *E(c)*. It is asymmetric matrix coinciding with *ECOUNT\_B* transposed.

ECOUNT\_D - matrix of expected counts *E(d)*.

Subcommands ADJUST, ABS and REVERSE are ignored when you request a matrix of observed or expected counts.

Sources:

* Deza, M.M, Deza, E. Encyclopedia of distances. Springer, 2009.
* Choi, S.-S., Cha, S.-H., Tappert, C.C. A Survey of binary similarity and distance measures // *Systemics, cybernatics and informatics*, 2010, 8(1), 43-48.
* Podani, J. Introduction to the exploration of multivariate biological data. Backhuys Publishers, 2000. Chapter 3.
* Romesburg, H.C. Cluster analysis for researchers. Lifetime Learning Publications, 1984.
* Alroy, J. A new twist on a very old binary similarity coefficient // *Ecology*, 2015, 96(2), 575–586.
* Jeff. simbin(mat1,mat2,type,mask) // MATLAB Central File Exchange (https://www.mathworks.com/matlabcentral/fileexchange/55190-simbin-mat1-mat2-type-mask), Retrieved October 18, 2020.
* Warrens, M.J. On association coefficients for 2×2 tables and properties that do not depend on the marginal distributions // *Psychometrika*, 2008, 73 (4), 777–789.
* Warrens, M.J. On the indeterminacy of resemblance measures for binary (presence/absence) data // *Journal of classification*, 2008, 25, 125-136.
* Goodman, L.A., Kruskal, W.H. Measures of Association for Cross Classifications. II: Further Discussion and References // *Journal of the American Statistical Association*, 1959, 54(285), 123-163.
* Eidenberger, H. Handbook of multimedia information retrieval. Atpress, 2012.
* Hubalek, Z. Coefficients of association and similarity, based on binary (presence-absence) data: an evaluation // *Biol. Rev*., 1982, 57, 669-689.

**On classification of binary similarity measures**. Many of or all similarity measures for binary data are often called matching coefficients. Two fundamental classes among them are **ordinal** measures and **nominal** measures; they differ by the treatment of “negative matches” count *d*, i.e. the frequency of match “absent attribute – absent attribute”, or “0-0”.

* **“Ordinal”** or **“one-pole”** similarities consider only count *a* (“1-1”), and not count *d* (“0-0”) to be the basis of affinity of the two vectors. For these measures, count *d* tells neither of similarity nor of dissimilarity, therefore it is absent in their numerator. That means that these measures treat values 1 vs 0 *asymmetrically*: 1 (“present”) is “more” than 0 (“absent”), – hence these measures interpret dichotomous data as ordinal.
* **“Nominal”** or **“two-pole”** similarities consider both *a* (“1-1”) and *d* (“0-0”) to be the basis of affinity of the two vectors. If two objects both lack some attribute, it is a reason for their similarity, like when they both have that attribute. Count *d* therefore is present in the numerator of such measure, along with count *a*. Thus, such measures regard values 1 vs 0 *symmetrically*: 1 (“present”) and 0 (“absent”) are two equal-rights categories, – hence these measures interpret dichotomous data as nominal. However, some of the nominal measures recognize that, mathematically, 1>0, and so they can take on not only positive but also negative values (in the fashion of a correlation).

**Ordinal** similarities include: *RR*, *JACCARD*, *DICE*, *SS2*, *K1*, *K2*, *MCCON*, *OCHIAI*, *SIMPSON*, *BRAUN*, *MOUNTF*, *FCOS*, *FORBES*, *PMI*, *ALROY*. **Nominal** similarities include: *SM*, *HAMANN*, *RT*, *SS1*, *SS3*, *SS4*, *SS5*, *GOWER*, *CHISQ*, *CHISQY*, *CC*, *PHI*, *PRPHI*, *DISPER*, *TETRACH*, *LAMBDA*, *D*, *Q*, *Y*, *DIGBY*, *GK1*, *MICHAEL*, *HD*, *TARWID*, *DENNIS*, *KAPPA*, *SPI*, *LOEV*, *COLE*, *MAXWELL*, *FLEISS*, *EYRAUD*, *GILBERT*. Measures *FAI2*, *FAI1*, *BUB2*, *BUB1* are in-between of ordinal and nominal because they recognize *d* to be the basis of affinity to a lesser degree than they recognize *a*.

Similarities *K2*, *MCCON*, *OCHIAI*, *DICE*, *SIMPSON*, *BRAUN*, *FCOS*, and also *SS4*, *SS5* may be called **conditional probabilities measures**, for their meaning are the proportions and (in classification tasks known as “Recall” and “Precision”), and also sometimes and . A number of similarities are tightly connected with **the chi-square statistic and the correlation**: *CHISQ*, *CHISQY*, *CC*, *PHI*, *PRPHI*, *TETRACH*, *DISPER*. Another row of measures is linked with the idea of **predictability and odds ratio**: *LAMBDA*, *D*, *Q*, *Y*, *DIGBY*.

**Relation between binary measures**. Linearly equivalent (Pearson r = 1) are measures within the following classes:

* *SM*, *SEUCLID*, *HAMANN*, *VARIANCE*
* *DICE*, *LW*
* *K2*, *MCCON*
* *CHORD*, *HELLINGER*
* *FAI2*, *FAI1*
* *BUB2*, *BUB1*
* *PRPHI*, *LOEV* (only in the domain of nonnegative values)

Measures *TETRACH*, *DIGBY* correlate tightly (Pearson r > 0.99). Measures *SS4*, *PHI*, *MAXWELL* – also.

Monotonically equivalent (Spearman rho = 1) are measures within the following classes:

* *SM*, *RT*, *SS1*, *EUCLID*
* *JACCARD*, *DICE*, *SS2*
* *OCHIAI*, *CHORD*
* *CHISQ*, *CC*
* *FORBES*, *PMI*, *TARWID*

Measures *TETRACH*, *Y*, *Q* correlate tightly (Spearman rho > 0.99).

**ADJUST**

By default and ADJUST=NONE, the macro outputs the observed-in-data value of the measure. S/c ADJUST allows to request adjusted (corrected) value of a similarity measure. This subcommand is only for similarities; it is not applicable to dissimilarities. Choose:

INDBOUND - correction for baseline independence, alias correction for association due to chance. The observed value of the measure gets rescaled so that will become equal to 0, if it is on the level of value expected under condition of no association between the vectors, that is, on the level characteristic for random unsystematic data.

INDUPPER - another variant of correction for baseline independence.

MARGPOLE - correction for the pole (extreme) value attainable under the given marginal distributions. The observed value of the measure gets rescaled so will become equal to 1, if it coincides with the maximal positive value attainable by the measure under the given marginal counts of the 2×2 table; and it will become equal to -1, if it coincides with the minimal negative value, attainable under the said conditions.

INDMARGPOLE - mixed correction: simultaneously for baseline independence and for the extreme value attainable under the given marginal distributions.

INDMARGMAX - another variant of alike mixed correction.

MARGPOLE, INDMARGPOLE, and INDMARGMAX are unavailable for measure *MOUNTF*. S/c ADJUST is ignored if MEASURE is counts (*COUNT*\_... or *ECOUNT*\_...).

All five methods of adjustment are rescaling of the observed similarity value by the common formula:

where is the observed value of similarity, and the other terms are defined below. If the formula numerator is positive, , and if negative, .

* With INDBOUND (correction for baseline independence):

is the similarity value computed from the table of expected (under rows–columns independence) counts

|  |  |
| --- | --- |
|  |  |
|  |  |

= upper bound of the general range of variation of the similarity, if , and = lower bound of the general range of variation of the similarity, if . The bounds, for every similarity, see in s/c MEASURE. *Note*: for measure *PMI* the lower bound (which is -∞) is assumed as -10.

* With INDUPPER (correction for baseline independence):

as with INDBOUND.

= upper bound of the general range of variation of the similarity.

* With MARGPOLE (correction for the attainable pole):

= 0 (and so sign of will coincide with sign of )

, if , and , if .

is the upper, and is the lower, range of variation of the similarity *under the given* marginal distributions which the two being compared vectors possess, i.e. under fixed marginal counts of the 2×2 table. Specifically:

is the similarity value computed from the table corresponding to co-sortedness of the vectors (that is, corresponding to their strongest positive association):

|  |  |
| --- | --- |
|  |  |
|  |  |

is the similarity value computed from the table corresponding to counter-sortedness of the vectors (that is, corresponding to their strongest negative association):

|  |  |
| --- | --- |
|  |  |
|  |  |

* With INDMARGPOLE (double correction):

as with INDBOUND

as with MARGPOLE.

* With INDMARGMAX (double correction):

as with INDBOUND

.

Tables of the aforementioned counts (*a’, b’, c’, d’* etc.) are returned by the macro in Output Viewer if the number of vectors in the analysis equals just two.

**ABS**

Optional subcommand abolishing negative sign in computed proximities (if the given proximity measure can be negative): ABS=YES. If ADJUST is specified, ABS is applied after it. S/c ABS is ignored if MEASURE is the counts (*COUNT*\_... or *ECOUNT*\_...).

**REVERSE**

This subcommand reverses computed proximities: similarities into dissimilarities, dissimilarities into similarities. Choose method:

NONE - (also default/unspecification) don’t reverse.

NEGSHIFT - change sign and then shift towards positive side. This is a universal linear method, it is suitable for all proximities. Similarities are transformed into dissimilarities by the formula , where is the highest off-diagonal similarity in the matrix, and on the diagonal of the result dissimilarity matrix 0 is set. Dissimilarities are transformed into similarities by the formula , where is the greatest dissimilarity in the matrix.

ONEMINUS - subtract from one. This method transforms similarities into dissimilarities by the formula and is suitable for similarities with values in the range [0, 1] or [-1, 1]. It transforms dissimilarities into similarities by the formula and is suitable for dissimilarities with values in the range [0, 1].

LAWCOS - this method is based on the law of cosines. It transforms similarities into dissimilarities by the formula and is suitable for similarities with values in the range [0, 1] or [-1, 1]. It transforms dissimilarities into similarities by the formula and is suitable for dissimilarities with values in the range [0, 1].

RECIP *k* - through inversion. This method transforms similarities into dissimilarities by the formula and is suitable for similarities with values in the range (0, 1]. It transforms dissimilarities into similarities by the formula and is suitable for any dissimilarities. *k* is a positive number. By default of the number *k* it is taken for 1.

REVERSE is applied after ADJUST and ABS. S/c REVERSE is ignored if MEASURE is the counts (*COUNT*\_... or *ECOUNT*\_...).

***Special regimes***

The macro ignores weighting of the dataset (however, it doesn’t take in the procedure cases with missing and nonpositive weights) and is not suited for splitting the dataset. The macro obeys case selection/filtering (commands FILTER, USE, SELECT IF, N OF CASES) and temporary (under TEMPORARY) transformations. If you use selection by commands SELECT IF or N OF CASES not under TEMPORARY, you’ve removed a portion of cases from the original dataset and the values of column CASENO\_ by output matrix are then the case numbers in that remaining dataset. In all other screening regimes, the values of CASENO\_ are the case numbers in the original dataset observed there prior your screening of cases.

# MACRO !KO\_TETRACH: TETRACHORIC CORRELATION COEFFICIENT

Version 1, Dec 2004. Tested on SPSS 11, 11.5, 13, 14.

!KO\_tetrach vars= v1 to v4 /\*Binary (0 and 1) variables between which correlations to compute, name-by-name and/or via “to” list

/missing= PAIRWISE /\*Exclude missing values listwise (LISTWISE) or pairwise (PAIRWISE, default).

Minimal specification is VARS

Макрос вычисляет и выдает как новый рабочий файл матрицу тетрахорических коэффициентов корреляции между входящими двоичными переменными. Тетрахорический коэффициент корреляции предназначен для таких дихотомических переменных, за которыми скрываются величины мерные, т.е. количественные и непрерывные (для которых обычно подходит интервальная или отношенческая шкала), причем имеющие в популяции нормальное распределение; дихотомичность же их представляется следствием огрубления измерительной шкалы до 2-х градаций. При таком (довольно смелом) допущении тетрахорический коэффициент есть предположительное значение коэффициента корреляции Пирсона, которое было бы, останься переменные мерными, не огрубленными до дихотомических.

Вычисление идет по «формуле косинуса», предложенной Пирсоном (Wherry R. J. Contributions to correlational analysis., 1984), дающей хорошее приближение для переменных не с очень перекошенным распределением (не более 90-95% наблюдений в одной из двух категорий). Макрос не вычисляет значимостей коэффициентов.

R = cos [ pi\*sqrt(BC) / [sqrt(AD)+sqrt(BC)] ]

A, B, C, D образуют частотную таблицу:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | VarY | |
|  |  | 0 | 1 |
| VarX | 0 | A | B |
| 1 | C | D |

***Подкоманды***

**VARS**

Укажите, списком и/или через “to”, входящие переменные. Переменные должны быть двоичной (0 и 1) кодировки. Если ваши дихотомические переменные иной кодировки, переделайте их в двоичные.

**MISSING**

Если есть пропущенные данные, то как исключать их:

PAIRWISE – исключить попарно, т.е. из каждой пары переменных отдельно (тж. по умолчинию или незаданию подкоманды).

LISTWISE – исключить списочно, т.е. из всех переменных, если хотя бы в одной наблюдение есть пропуск.

***Особые режимы***

Макрос игнорирует взвешивание файла и не рассчитан на расщепленность файла.

# MACRO !KO\_BISER: BISERIAL CORRELATION COEFFICIENT

Version 1, Aug 2005. Tested on SPSS 11, 11.5, 13, 14.

!KO\_biser scale= **v1 to v5** /\*Scale variables, name-by-name or/and via “to”

/binar= *b1 to b3* /\*Binary (0 and 1) variables assumed coming initially

/\*from normal disribution, name-by-name or/and via “to”

/missing= /\*Exclude missing values listwise (LISTWISE) (LISTWISE)

/\*pairwise (PAIRWISE, default)

/alpha= /\*Optional: return only coef-s significant at this 2-tailed level

/\*(turn nonsignificant to zeros); if omit all coef-s and their sig-s are returned.

Minimal specification SCALE, BINAR.

The macro computes, and outputs in Output Viewer and in a new unnamed dataset, biserial correlation coefficients between scale and binary variables, as well as their significances in Output Viewer. There is an option to show, instead of coefficient significances, just significant coefficients on a specified alpha level.

Biserial correlation coefficient is meant to correlate a scale variable (i.e., a quantitative continuous one, measured usually by interval or ratio scale level) with a dichotomous variable under which a scale quantity hides, the latter having normal distribution in population; while the dichotomy of that variable is seen to be the result of coarsening its measuring scale to 2 levels. If this (quite bold) assumption is made, then the biserial coefficient is a suppositional (inferred) value of Pearson correlation coefficient that would be obtained, stay the second variable scale, not coarsened into the dichotomy. Formulas (Wherry R. J. Contributions to correlational analysis., 1984):

Rbis = (M1-M0)/S \* PQ/H

M1 and M0 – means of the scale variable in the two levels of the dichotomous variable;

S – st. deviation in the scale variable;

P and Q – the proportions in the distribution of the dichotomous variable, P+Q=1;

H – ordinate (probability density) of the st. normal curve in the point of division the area under the curve into P and Q.

Approximate significance (may use if the distribution of the dichotomous variable is not too skewed: P or Q <= 0.9 - 0.95):

Z = Rbis\*sqrt(N) / (sqrt(PQ)/H – Rbis^2)

Z substitutes into the st. normal distribution to obtain 2-sided *p*-value.

Computed biserial coefficient may in rare cases exceed 1 by absolute value. Theoretically correlation coefficient cannot be above 1, and the macro warns if coefficients absolutely higher than 1 were obtained. Such coefficients are a symptom (one of) that the assumption of underlying normality of the dichotomous variable is false.

***Subcommands***

**SCALE**

Scale variables (continuous or with scale with enough number of levels), name-by-name and/or via “to”. Variable names – up to 8 bytes long. There must be no value: -999 in the data.

**BINAR**

Dichotomous variables in binary (0 and 1) coding, name-by-name and/or via “to”. If your dichotomous variables are coded elsewise, redo them into binary. Variable names – up to 8 bytes long.

**MISSING**

If there are missing data, how to exclude them:

PAIRWISE - (default/unspecifying) delete cases pairwise, i.e., from each pair of (being correlated) variables separately.

LISTWISE - delete cases listwise, i.e., from all the variables, if at least on one of them the case is missing.

**ALPHA**

By default/unspecifying of the subcommand, the macro outputs all computed coefficients and their 2-sided significances. But instead, you can specify a critical significance level (alpha), - then all coefficients nonsignificant at his level will appear zeroed in the results (including coefficients output as a dataset), and there will be no significances in the results.

***Special regimes***

The macro ignores weighting of the dataset (however, it doesn’t take in the procedure cases with missing and nonpositive weights) and is not suited for splitting the dataset. The macro obeys case selection/filtering (commands FILTER, USE, SELECT IF, N OF CASES) and temporary (under TEMPORARY) transformations.

# MACRO !KO\_ACORRD: AUTOCORRELATIVE DISTANCE

Version 1, Feb 2006. Tested on SPSS 11, 11.5, 13, 14.

!KO\_acorrd vars= *v1 to v5* /\*List of numeric variables (may use “to”); these are items

/\*between which distances to compute

/negat= ASIS /\*Returned negative distances: accept them (ASIS);

/\*discard sign (ABS, default); turn to zero (ZERO)

/missing= LINT /\*Treatment of missing values:

/\*DELETE - delete cases listwise (default);

/\*LAG - replace missing value by value of the nearest preceding

/\*valid case; MEAN - replace missing value by the mean of 2 nearest

/\*surrounding it valid cases; LINT - replace missing value

/\* by the linear interpolation of 2 nearest surrounding it valid cases.

Minimal specification VARS.

This distance confers importance to differences which are systematic in the sense of (1-lag) autocorrelation, that is, they last over transition from case to next case (if you are comparing between variables) or from variable to nest variable (if you are comparing between cases). This macro compares between *variables*, so if your goal is to compare cases, transpose the data first (menu Data – Transpose). The macro outputs the matrix of distances as a new unnamed dataset.

For the autocorrelative distance, it is important the order in what cases go in the dataset, because the distance’s attitude to the sequence of cases is as to a “run” or a “time series”.

N

DXY = sum [ (X-Y)i \* (X-Y)i-1 ], X, Y – values of the two being compared variables in the current case i or in i-1.

i=2

As you can see from the formula, the autocorrelative distance is the unnormalized *coefficient of autocorrelation of differences* between the two series, X and Y, and therefore it can take on negative values too. A negative autocorrelation corresponds to the “switching” systematicity of differences: in one case X>Y, in the next case X<Y, then back again, and so on. This pattern is seldom encountered in the data gathered by social sciences. In those, the “enduring” systematicity of differences prevails: X>Y (or X<Y) at a cut of considerable length in the series under study (such as: time points, price levels, etc.); the said systematicity corresponds to positive autocorrelation. The greater is the difference and the “longer” it lingers on along the series, the greater will be autocorrelative distance. Large differences, if they are not steady, will give distance values close to 0, exactly like almost absent differences.

***Subcommands***

**VARS**

Indicate name-by-name and/or via “to” numeric quantitative (not nominal) variables distances between which interest you. Names of the variables - up to 8 bytes long. There should be no value: exactly -999 in their data. If magnitudes in the input data are large while cases are many (thousands), distances over the output matrix may get also very large. If you find that inconvenient, divide the data beforehand by some constant.

**MISSING**

If the first case in the dataset has at least one missing value, the macro excludes this case. The same pertains to the last case in the dataset, except when MISSING=LAG, under which it is not deleted but replaced. For all other cases with missing data, you may request what to do:

DELETE - (default/unspecifying) delete the case, if at least in one of the variables it is missing (system or user-missing).

LAG - replace the missing value with the nearest preceding (above in the dataset) valid value.

MEAN - replace the missing value with the mean of the pair of nearest, above and below in the dataset, valid values.

LINT - replace the missing value with the linear interpolation between the pair of nearest, above and below in the dataset, valid values. Interpolation gives another effect, than MEAN, in adjacent missing cases.

**NEGAT**

Negative distances, if such appear, can be modified before the matrix is output as a dataset.

ASIS - leave all distances as they are.

ABS - (default/unspecifying) abolish negative sign of distances.

MEAN - turn negative distances into 0.

If negative autocorrelation of differences means the presence of dissimilarity, like positive autocorrelation does, use ABS. If it is an artefact for you or tells of no dissimilarity, use ZERO.

***Special regimes***

The macro ignores weighting of the dataset (however, it doesn’t take in the procedure cases with missing and nonpositive weights) and is not suited for splitting the dataset. The macro obeys case selection/filtering (commands FILTER, USE, SELECT IF, N OF CASES) and temporary (under TEMPORARY) transformations.

# MACRO !KO\_RESCR: RESCALED PEARSON CORRELATION COEFFICIENT

Version 1, Dec 2009. Tested on SPSS 13, 15, 17, 26.

!KO\_rescr vars= *v1 to v10* /\*List of numeric variables (may write via to) between which

/\*correlations to compute

/rescale= POLE /\*Rescale to the nearest limit (POLE, default) or

/\*to range bw the limits (RANGE)

/eigen= YES /\*Control that matrix of rescaled r be Gramian: YES (default) or NO

/save= *'d:\exercise\newdata.sav'* /\*Optionally, when EIGEN=YES: fit variables for

/\*rescaled r: external filename or declared dataset name

/id= *id* /\*Optionally: case identificator variable

/print= YES /\*Show correlations in Output Viewer: YES or NO (default).

Minimal specification VARS.

The macro computes Pearson *r* between input variables and rescales its value relative the limit or limits that *r* could really attain in conditions of the data observed: in conditions of the variables with precisely such empirical marginal distributions as the data have.

The macro outputs the matrix of rescaled *r* in a new unnamed dataset. In the Viewer Output, the macro informs on the average (between all *r*) absolute size of difference of rescaled *r* from their corresponding usual (i.e., pre-rescaled) *r*. There are more options in the macro (see s/c EIGEN and SAVE).

Pearson correlation coefficient can theoretically vary from -1 to +1. However, in real data the empirical range beyond which *r* between two concrete variables cannot pass – is usually narrower than the theoretical one. Linear correlation between X and Y has a chance to reach +1 when and only when the shape of distribution is completely identical in X and Y (or, in other words, X and Y distributions differ no more than linearly). Otherwise, the upper limit for *r* will be below +1. Analogously, *r* can reach -1 when and only when the shape of distribution is completely identical in X and -Y. Otherwise the lower limit for *r* will be above -1. In order the correlation to have simultaneously the limits upper +1 and lower -1, the two distributions must be not only identical in shape, but precisely symmetric in shape. Real variables, quantitative or discrete features (including Likert and binary data), have shape of distribution often different from one another and often asymmetric; therefore, the empirical range for *r* is narrowed to this or that extent.

Large narrowing of the empirical range of *r* means that strength of association between the two variables is underestimated by Pearson correlation coefficient, and is underestimated the more the stronger the association is. Because strong relation is then inevitably nonlinear; but *r* only “skims” the linear portion of the relation. If a researcher wants in such a situation to “skim” more of association by a coefficient applied, they should think of applying some nonlinear coefficient of correlation, for example, Spearman *rho* nonparametric coefficient. But there is yet another solution option: to rescale the observed Pearson *r* relative its empirical limits. Say, if there observed *r* = 0.4, while for the given two observed variables *r* cannot become above 0.95, then *r* gets corrected: 0.4/0.95 = 0.42. This is rescaling of *r* to one, the nearest, limit (pole). Or one may rescale *r* to both limits together, the positive and the negative; but that is less common. The macro offers this and that mode of rescaling, to choose.

***Subcommands***

**VARS**

Indicate name-by-name and/or via “to” numeric variables correlations between which interest you. Names of the variables - up to 8 bytes long. The macro deletes missing values listwise: a case is removed totally if it is missing in at least one of VARS.

**RESCALE**

Way of rescaling the computed correlation coefficient:

POLE - (default/unspecifying) the observed (usual) value of *r* is rescaled to its closer empirical limit. In other words, it is the rescaling from range {0 : +1} (if the given *r* value is positive) or from {0 : -1} (if the given *r* value is negative) to range {0 : the limit of the same sign as the given *r* value}. Rescaled nonzero *r* will always be greater in abs. magnitude than the usual *r*. Zero *r* will remain zero.

RANGE - the observed value of *r* is rescaled to both empirical limits together, that is, from range {-1 : +1} to range {negative limit : positive limit}, so that, as the result, *r* will occupy in the latter the same relative position that it occupied in the range {-1 : +1}. Rescaled *r* may, in abs. magnitude, be greater or lesser than the usual *r*. Close to zero *r* may become zero and even change sign, zero *r* may become nonzero.

**EIGEN**

By default/unspecifying the subcommand and with EIGEN=YES, the macro watches it the matrix of rescaled *r* to be positive (semi)definite (aka Gramian), i.e., will not produce negative eigenvalues when decomposed; and if it turns out not so, the macro will be lessening (unwinding) the difference magnitude of the rescaled *r* from the usual *r* until the matrix becomes Gramian. The difference magnitude will be lessened for all matrix elements by the same number of times. (Some forms of analysis, for example factor analysis, require that the input matrix be positively definite or semidefinite.)

Matrix of rescaled *r* appears not positive (semi)definite (not Gramian) typically with strongly discrete input data, for instance, dichotomous.

If among input variables VARS there complete collinearity is observed, the matrix of initial *r* is singular; then the matrix of rescaled *r* is almost surely non-Gramian and, moreover, EIGEN subcommand is likely not to cope. So input data without collinearities.

With EIGEN=NO, the macro always leaves *r* that what they turned out at the rescaling. With EIGEN=NO, s/c SAVE and ID are ignored.

**SAVE**

This subcommand is inactive with EIGEN=NO. You may, by specifying path/name of an external .SAV file or name of an internal existing or declared dataset, request that the macro saves there the input variables, fitted (modified) by it to them reproducing the final rescaled *r*. I.e., usual *r* coefficients computed on those fitted variables will equal the rescaled *r* (the ones output in the new unnamed dataset). Making variables to fit is done by the “principal component method with procrustes post-rotation”[[4]](#footnote-4). This method tries to alter variable values gently.

The variables will be saved in standardized form (mean = 0, st. dev. = 1); you may afterwards rescale them yourself as you like[[5]](#footnote-5).

**ID**

The subcommand makes sense if SAVE is specified. Indicate one variable identifying cases in the input data. Since cases with missings, if any, will be omitted by the macro, an ID variable will allow you to merge subsequently the variables saved by s/c SAVE with the dataset input to the macro.

**PRINT**

PRINT=YES prints the matrix of *r* of input variables and the matrix of rescaled *r* to Output Viewer. By default, PRINT=NO.

***Special regimes***

The macro ignores weighting of the dataset (however, it doesn’t take in the procedure cases with missing and nonpositive weights) and is not suited for splitting the dataset. The macro obeys case selection/filtering (commands FILTER, USE, SELECT IF, N OF CASES) and temporary (under TEMPORARY) transformations.

1. In literature, there also is found a version of the Hellinger distance formula, where instead of multiplier 2 under the root, multiplier ½ is used. [↑](#footnote-ref-1)
2. SPSS Statistics command PROXIMITIES computes this chi-square distance too, and for any nonnegative data, for example, raw frequencies. The macro does it for proportions only. [↑](#footnote-ref-2)
3. Some similarity measures may yield a diagonal 0 if the vector is a constant. [↑](#footnote-ref-3)
4. См. функцию !KO\_TOCOV из коллекции “MATRIX - END MATRIX functions”. [↑](#footnote-ref-4)
5. Для этой цели удобна функция !KO\_RESCALE из коллекции “MATRIX - END MATRIX functions”. [↑](#footnote-ref-5)