***Series Response tools***

SPSS macros by Kirill Orlov

kior@akado.ru, ttnphns@gmail.com

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

All rights reserved

*Инструменты для серии пунктов.* Собрание макросов для «простого матричного вопроса», т.е. серии переменных с общим пулом альтернативных вариантов ответа (Single response series, SRS), - например набора пунктов, каждый из которых оценен по балльной шкале или проранжирован. Один макрос предназначен для данных ранжирования и переводит переменные в категориальный набор множественного ответа или обратно. Другой макрос предназначен для более общих задач перевода значений и переменных друг в друга и для обсчета повторяющихся значений. Третий макрос предназначен для ситуации, когда респонденты оценивали не все пункты, а те только, которые они предварительно выбрали, набивка же была сделана уплотненным (ускоренным) способом.

* Если был вопрос на ранжирование или выбор-и-ранжирование, то как бы он ни был набит – в виде серии пунктов или в виде категориального набора множественного ответа, [!KO\_RANKREV](#_МАКРОС_!RANKREV:_ПЕРЕВОД) перекодирует одну набивку в противоположную.
* Если перед вами стоит более общая задача типа «сделать из значений переменные» или «сделать из переменных значения», используйте [!KO\_VALVARS](#_МАКРОС_!KO_VALVARS:_ПЕРЕВОД_1). Этот макрос полезен и для подсчета некоторых статистик по-горизонтали, особенно в условиях запрета на учет части ячеек данных.
* Ручная набивка вопроса типа «выберите и оцените выбранные пункты» является проблемой (приходится искать нужную переменную, в которую набить оценку). Макрос [!KO\_SRSREF](#_МАКРОС_!SRSREF:_РАЗВЕРТКА) решает ее. Вы можете набивать оценки подряд («уплотненно») – макрос развернет эту набивку в правильный набор переменных.

**Серия пунктов на один ответ** (Single response series, SRS) это набор пунктов или подвопросов с общим банком вариантов ответа; на каждый пункт можно дать только один ответ. Другое название для такого набора – простой матричный вопрос (simple matrix question). Типичным примером являются пункты, оцениваемые по шкале (вроде ликертской, осгудской и пр.) либо просто «Да – Нет». Другим примером могут быть вопросы на ранжирование, когда надо распределить пункты по порядковым местам. Переменные в SRS-наборе - это пункты, а значения это коды ответов, шкальные оценки или ранги.

# МАКРОС !KO\_RANKREV: ПЕРЕВОД ПЕРЕМЕННЫХ С РАНГАМИ В ПЕРЕМЕННЫЕ-РАНГИ ИЛИ ОБРАТНО

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

!KO\_rankrev vars= *v1 v2 v3 v4* /\*Поименный список набора переменных для перекодировки

/capnum= *w\_ 3* /\*Приставка в имена новых переменных;

/\*после нее можно указать их кол-во (равное верхнему значению в VARS);

/\*Если capnum не задать, то сами VARS будут перекодированы

/filler= /\*Опционально: значение в незаполненные ячейки в новых переменных

/\*(оно будет user-missing).

Минимум надо задать VARS.

Данные вопроса на ранжирование (ranking: распределить некие пункты или варианты по местам: 1-е, 2-е, и т.д.) могут быть набиты двояко. В одном случае – как серия пунктов (SRS-набор), т.е. переменные отвечают пунктам, а данные в них есть занятые места, ранги; в другом случае – как категориальный набор множественного ответа (MRC), где переменные есть «попытка» ответа по порядку (а в данном случае – приписанный ранг), а данные в переменных это коды пунктов. К примеру, если были пункты (какие-то стимулы), пронумерованные как **1, 2, 3**, и респондент расставил их, соответственно, по местам *3, 1, 2,* то набивка в виде SRS будет:

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

а в виде MRC будет:

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

В первом случае ранги (данные) – это значения, а во втором случае они – это переменные.

Набивка в виде MRC экономнее в ситуации, когда вопрос требует проранжировать не все пункты, а только те, которые респондент выберет (это т. наз. вопрос на множественный ответ с ранжированием), – потому что тогда MRC может состоять из меньшего числа переменных, чем существует пунктов. С точки зрения анализа данных, MRC-набор можно подвергнуть табуляции как набор множественного ответа, т.е. без учета рангов, или можно анализировать распределение пунктов внутри 1-го места, 2-го места, и т.д. SRS-набор, с другой стороны, позволяет анализировать переменные как порядковые или мерные данные, например посмотреть средний ранг, получившийся у каждого пункта.

Данный макрос переводит одно строение данных «вопроса на ранжирование» в другое, т.е. SRS-набор в MRC-набор или же наоборот. Вы можете затребовать создать новые переменные или изменить сами входящие переменные. Значения во входящих переменных должны быть порядковыми числами 1, 2, 3…, причем значение должно встречаться у наблюдения не более раза[[1]](#footnote-1). (Если это не так, то используйте более универсальный макрос !KO\_VALVARS.) Макрос !KO\_RANKREV (“ranking reverse”) поступает так: если в i-й входящей переменной, как они расположены в VARS, значение наблюдения = j, то в j-ой выходящей переменной оно будет = i.

***Подкоманды***

**VARS**

Укажите поименно набор переменных, которые нужно перекодировать в «противоположное» строение. Это может быть MRC-набор или его часть, либо SRS-набор или его часть. Имена входящих переменных могут быть любыми. Значения входящих переменных должны быть натуральными числами 1, 2,…. Часть значений этого ряда может реально в данных отсутствовать. Кроме натуральных чисел, все иные значения, а также значения-пропуски, макрос игнорирует. Значения, превосходящие верхний предел, указанный в п/к CAPNUM, игнорируются; по умолчанию этот предел равен числу переменных VARS.

Значения, которые получатся в выходящих переменных, это порядковые номера входящих переменных VARS. Выходящие переменные получают мерный (scale) мерительный уровень; хотя в действительности они или порядковые (если они суть SRS), или номинальные (если они суть MRC).

**CAPNUM**

Укажите приставку в имена выходящих переменных, ее можно взять в кавычки/апострофы (они обязательны, если вместе с приставкой вы используете разделитель-точку: ‘var.’). Имена выходящих переменных будут образованы из приставки и порядковых чисел 1, 2,…, которые суть значения во входящих переменных. Вы можете указать после приставки «верхний предел» – целое число, равное максимальному значению, существующему или интересующему вас во входящих переменных, - столько и будет создано выходящих переменных. По умолчанию такого числа после приставки макрос берет в качестве него число входящих переменных.

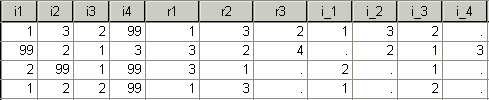
Переменные создаются новыми. Переменных с такими именами должно не быть в вашем массиве. (Если какие-то из создаваемых переменных уже существуют в массиве, SPSS сообщит об ошибке. Тогда проверьте эти переменные – были они изменены макросом или нет.)

Вы можете опустить задание CAPNUM или умолчать подкоманду. В этом случае выходящими будут сами входящие переменные: макрос перекодирует их, полностью обновя. Ярлыки значений и user-missing спецификации будут удалены. Ярлыки переменных макрос оставляет как есть. Когда CAPNUM не задано, «верхний предел», упомянутый выше, полагается дефолтным.

ПРИМЕР 1. В вопросе на ранжирование предлагалось 4 предмета (пункта). Респонденты должны были выбрать из них не более 3-х и проранжировать по важности (1-е место, 2-е, 3-е). Исходно данные были набиты как SRS-набор, т.е. переменные *I1 – I4* это пункты, а значения в них – ранги.

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

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



* Первая команда переводит SRS в MRC, состоящий из переменных *R1, R2, R3*, которых три: это число было заказано, т.к. разрешено было проранжировать до трех предметов. *R1* это «ранг 1», *R2* «ранг 2», *R3* «ранг 3». Значения в них – это предметы от 1 до 4, в соответствие входящим переменным. У наблюдения 4 ранг 2 встречался дважды, - учтено только одно, второе его появление.
* Вторая команда делает обратный перевод MRC в SRS, в новые переменные *I\_1, I\_2, I\_3, I\_4*. Число 4 указывает, что создать нужно 4 переменных, т.к. было 4 предмета (с кодами от 1 до 4 в MRC).

**FILLER**

По умолчанию макрос оставляет в выходящих переменных пустые ячеки пустыми (system-missing). Здесь вы можете указать код-наполнитель (любое число), которое вставить туда. Ему будет придан статус пропущенного значения (user-missing). Макрос *не* вставляет наполнитель в наблюдения, которые пропущены (user- или system-missing) во *всех* переменных VARS; такие наблюдения выходят как system-missing.

***Особые режимы***

Макрос не реагирует на взвешенность, фильтрованность (FILTER, USE), расщепленность массива данных. Он не слушается временных (под TEMPORARY) преобразований.

# МАКРОС !KO\_VALVARS: ПЕРЕВОД ЗНАЧЕНИЙ И ПЕРЕМЕННЫХ ДРУГ В ДРУГА

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

!KO\_valvars vars= *v1 v2 v3 v4* /\*Входящие числовые переменные, поименно

/values= *1 to 8* EXCEPT *3 5* /\*Список значений (неотриц числа, или поименно,

/\*или диапазоном ч-з to) интересующих вас в VARS; после диапазона можно

/\*перечислить, после сл EXCEPT, значения, к-рые требуется исключить из него

/newvars= 'w\_' /\*Имена новых переменных: поименно (сколько VALUES) или единственная приставка

/newvals= EXTRACT /\*Значения в новых переменных: 1) перечислить (сколько VARS): неотриц числа

/\*и/или имена переменных;

/\*2)ORDINAL - породить целые 1,2,...,число VARS;

/\*3)EXTRACT - взять из имен VARS; 4) ALLONE - все значения 1

/filler= 99 /\*Опционально: значение в незаполненные ячейки в новых

/\*переменных (оно будет user-missing)

/miss= SET /\*Если в созданных переменных нужны пропущенные наблюдения: делать его sysmis,

/\*если 1) оно missing во всех VARS (SET);

/\*2) если в данной переменной (не обязат из VARS) его значение missing

/\*(VAR имя переменной)

/redupl= MINGAP /\*Если в VARS значение (из VALUES) может встречаться более раза,

/\*то какое учитывать:

/\*FIRST, LAST (п/у), RANDOM, MAX, MIN, обсчитать все:

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

Минимум надо задать VARS, VALUES, NEWVARS, NEWVALS.

Макрос предназначен для задач, которые в общих чертах можно понять как «перевод значений в переменные, а переменных в значения». Скажем, когда переменные это пункты и значения это ответы, можно перевести в такое строение, когда переменные это ответы, а значения это пункты. Или же перевести в обратном направлении. Это более общий, чем !KO\_RANKREV, макрос, рассчитанный не только на ранговые данные и предлагающий больше возможностей. Он применим также для вычисления некоторых статистик по горизонтали, т.е. внутри наблюдений. Макрос создает только новые переменные.

Сущность работы макроса состоит в следующем.

Пусть есть *p* входящих переменных VARS: *VAR1, VAR2,…, VARp,*

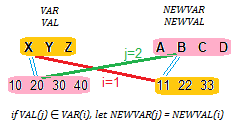
в которых вас интересует список *k* значений VALUES: *val1, val2,…, valk.*

Вы задаете в соответствие входящим переменным *p* новых значений[[2]](#footnote-2) NEWVALS *newval1, newval2*,…, *newvalp,*

а в соответствие входящим значениям – *k* новых (еще не существующих) переменных NEWVARS: *NEWVAR1, NEWVAR2,…, NEWVARk.*

Макрос создает эти *k* переменных – так, что если у респондента (наблюдения) значение *valj* принадлежит переменной *VARi*, то переменная *NEWVARj* получает значение *newvali* (**Рис. 1**). Значения VALUES входят в имена переменных NEWVARS.

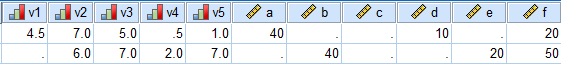
Если у респондента во входящих переменных есть повторяющиеся значения (т.е. *valj* принадлежит нескольким переменным VARS), то макрос обладает рядом возможностей их учета/обсчета (см. п/к REDUPL).

****

**Рис. 1.** Переменные это столбцы массива, а значения это данные в наблюдении (строке). Макрос делает так, что входящим переменным соответствуют выходящие значения, а входящим значениям соответствуют выходящие переменные.

ПРИМЕР 1.

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



* В переменных *V1-V5* (пять) нас интересуют значения .5, 2, 3, 4.5, 6, 7 (шесть). Новые переменные *A-F* (шесть) пусть будут иметь возможные значения 10, 20, 30, 40 ,50 (пять).
* Рассмотрим 1-е наблюдение. Значение .5 (оно 1-е в списке) находится в переменной *V4* (она 4-я в списке), поэтому в 1-й новой переменной (*A*) будет 4-е новое значение (40).
* Значений 2, 3 нет в 1-м наблюдении. Значение 4.5 (оно 4-е в списке) находится в переменной *V1* (она 1-я в списке), поэтому в 4-й новой переменной (*D*) будет 1-е новое значение (10).
* Значения 6 нет в 1-м наблюдении. Значение 7 (оно 6-е в списке) находится в переменной *V2* (она 2-я в списке), поэтому в 6-й новой переменной (*F*) будет 2-е новое значение (20).
* Во 2-м наблюдении значение 7 встречено более раза, т.е. находится в разных переменных. По умолчанию п/к REDUPL макрос учитывает последнюю встречу значения. Поэтому новая переменная *F* получает новое значение 50.

ПРИМЕР 2. Этот пример воспроизведет работу макроса [!KO\_RANKREV](#_МАКРОС_!RANKREV:_ПЕРЕВОД), показанную там в ПРИМЕРЕ 1.

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

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

* Первая команда переводит набор переменных *I1, I2, I3, I4*, в которых нас интересуют значения 1, 2, 3, в набор, соответственно, переменных *R1, R2, R3*. Если, положим, *I2*=1 (одно из VALUES), то на выходе будет *R1*=2 (порядковый номер *I2* в списке VARS).
* Вторая команда делает обратный перевод. Она переводит *R1, R2, R3,* в которых нас интересуют значения 1, 2, 3, 4, в, соответственно, *I\_1, I\_2, I\_3, I\_4*. Если, положим, *R1*=2 (одно из VALUES), то на выходе будет *I2*=1 (порядковый номер *R1* в списке VARS).

***Подкоманды***

**VARS**

Укажите поименно набор входящих переменных (минимум две). Это могут быть любые числовые переменные.

**VALUES**

Список интересующих валидных значений в VARS, одно или несколько неотрицательных чисел[[3]](#footnote-3). Задать список можно или перечислением, или диапазоном через “to”: min to max; но не смешанно. В первом случае значения не обязаны быть целыми числами. Когда же они заданы диапазоном, макрос формирует список как целые числа от min до max, которые должны быть целыми.

Вставкой ключевого слова EXCEPT после диапазона можно задать перечислением список значений, которые вы хотите выпустить из диапазона. Например, если VALUES= *1 to 5* EXCEPT *2 4*, то список значений, который сформирует макрос, будет *1 3 5*. Если после слова EXCEPT кодов нет, то само слово игнорируется.

**NEWVARS**

Это список выходящих, новых переменных. Они должны несовпадать именами с какими-л. существующими переменными массива данных[[4]](#footnote-4). Число переменных в списке должно равняться числу значений VALUES, т.к. эти два списка соответственны элементами. Задайте поименный список имен новых переменных.

Вместо поименного списка можно указать единственную приставку, которую можно взять в кавычки/апострофы (они обязательны, если приставка заканчивается разделителем-точкой, “var.”). Тогда имена новых переменных будут состоять из этой приставки и самих VALUES в качестве окончаний.

Выходящие переменные получают мерный (scale) мерительный уровень. Вы потом сами установите уровень, который желаете.

**NEWVALS**

Это список значений новых переменных. Он и список VARS соответственны элементами. Можно задать его по-разному:

*Список значений* *и/или переменных* - перечислите неотрицательные числа в количестве переменных VARS, в соответствие им. Числа не обязаны быть все разными. Можно указывать и имена существующих в массиве числовых переменных, которые содержат необходимые значения (не обязательно неотрицательные). Задание списка переменными значит, что для каждого наблюдения возможен свой список значений. Пропущенные значения в этих переменных, если есть, не будут переноситься в создаваемые макросом новые переменные. Среди переменных могут фигурировать и переменные VARS.

ORDINAL - макрос формирует список автоматически как натуральные числа 1, 2, …, число входящих переменных.

EXTRACT - макрос пытается экстрагировать значения из имен входящих переменных. Это могут быть только целые числа. Входящие переменные должны оканчиваться на эти числа-значения. Например, переменная с именем *VAR2* даст значение 2. Переменная с именем *MRC3.2* также даст значение 2. Макрос берет целое число из хвоста имени.

ALLONE - это то же, что список значений из одних единиц. Имеет смыл вместе с REDUPL=SUM для подсчета одинаковых значений во входящих данных.

**FILLER**

Необязательная подкоманда. По умолчанию, макрос оставляет пустыми (system-missing) ячеки, оставшиеся незаполненными в выходящих переменных. Здесь вы можете указать код-наполнитель, любое число, которое вставить туда. Ему будет придан статус пропущенного значения (user-missing).

**MISS**

Подкоманда для учета/создания пропущенных значений. По умолчанию/незаданию и при MISS=NO пустых ячеек (system-missing) в выходящих переменных не будет, если вы задали FILLER. MISS позволяет сделать system-missing некоторые наблюдения. MISS делает наблюдение system-missing *во всех* выходящих переменных. Укажите одно из двух условий:

SET - наблюдение будет system-missing, если оно имеет пропущенные значения (user-missing или system-missing) *во всех* входящих переменных VARS.

VAR *имя переменной* - наблюдение будет system-missing, если оно имеет пропущенное значение (user-missing или system-missing) в этой конкретной указанной переменной (она может быть любой переменной массива данных).

MISS делает наблюдения system-missing *после* того, как выходящие переменные с их данными созданы.

**REDUPL**

Если значение из VALUES встречается у респондента (наблюдения массива) более одного раза во входящих переменных VARS, когда последних несколько, то возникает проблема, какое именно из них учитывать: ведь соответственные значения NEWVALS у них (у этих VARS) могут быть разные. В подкоманде REDUPL указывается, как учитывать повторяющиеся в VARS значения.

Следующие опции учитывают *одно* из одинаковых значений – значение, принадлежащее i-й переменной в VARS (следовательно, берется i-е значение NEWVALS):

LAST - (тж. по умолчанию/незаданию) учесть последнюю в череде VARS встречу значения.

FIRST - учесть первую в череде VARS встречу значения.

RANDOM - выбрать из одинаковых значений одно случайно[[5]](#footnote-5).

MIN - учесть значение той переменной VARS, для которой меньшее значение NEWVALS.

MAX - учесть значение той переменной VARS, для которой большее значение NEWVALS.

Следующие опции учитывают *все* встречи значения и берут – в качестве значения NEWVALS – сводную статистику из всех соответствующих им значений NEWVALS:

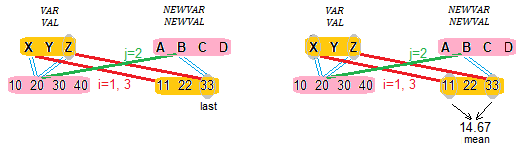
MEAN - вычисляется средняя их значений NEWVALS.

SUM - вычисляется сумма их значений NEWVALS.

RANGE - вычисляется диапазон их значений NEWVALS. Если значение встречается только раз, оно даст диапазон 0.

MINGAP - вычисляется минимальный разрыв (разница) между их значениями NEWVALS. Если значение встречается только раз, оно даст разрыв 0.

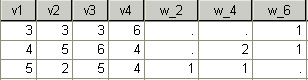
MAXGAP - вычисляется максимальный разрыв (разница) между их значениями NEWVALS, если последние расположить по возрастающей и рассматривать там разницы между соседями. Если значение встречается только раз, оно даст разрыв 0.



**Рис. 2.** REDUPL=LAST и REDUPL=MEAN. Значение 20 встречается в VARS дважды: в *X* и *Z*; поэтому надо выбирать между NEWVALS 11 и 33, либо вычислять их сводную статистику.

ПРИМЕР 3. Подсчет значений горизонтально. Макрос играет роль SPSS-команды Count.

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



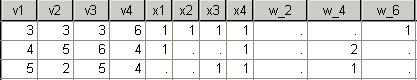
Из VALUES следует, что интересующие значения во входящих переменных – 2, 4, 6, которым и отвечают выходящие переменные с приставкой *W\_*.

REDUPL заказывает накапливать сумму значений NEWVALS при каждой встрече интересующего значения. Поскольку NEWVALS=ALLONE (что = 1 1 1 1), получается подсчет интересующих значений.

ПРИМЕР 4. Тот же пример подсчета значений, но с частичным запретом на подсчет.

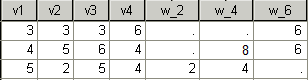
!KO\_valvars vars= v1 v2 v3 v4 /values= 2 to 6 EXCEPT 3 5 /newvars= 'w\_' /newvals= x1 x2 x3 x4

/redupl= SUM.



В подкоманде NEWVALS вместо ALLONE указаны переменные *X1–X4*, которые содержат единицы не везде: пустые ячейки налагают запрет на учет соответствующих им ячеек в переменных *V1-V4*. Таким образом, подсчитаны не все появления значений 2, 4, 6 в *V1–V4*.

ПРИМЕР 5. Вычисление горизонтальной суммы одинаковых значений.



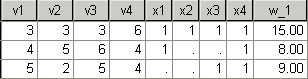
!KO\_valvars vars= v1 v2 v3 v4 /values= 2 to 6 EXCEPT 3 5 /newvars= 'w\_' /newvals= v1 v2 v3 v4

/redupl= SUM.

В качестве NEWVALS указаны сами переменные VARS. Т.к. REDUPL=SUM, выходящие переменные *W\_2, W\_4, W\_6* будут содержать суммацию повторов, соответственно, значений 2, 4, 6.

ПРИМЕР 6. Вычисление горизонтальной статистики (например суммы значений) с частичным запретом.

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



В фокусе интереса – переменные *V1 – V4*, которые на сей раз упомянуты как NEWVALS. Единицы содержатся в переменных *X1 – X4*, указанных как VARS, и в качестве VALUES указано значение 1. Таким образом, макрос вычисляет горизонтальную сумму (REDUPL=SUM) *V1 – V4*.

Т.к. *X1–X4* имеют пустые ячейки, то соответствующие им ячейки в *V1 – V4* табуированы и не участвуют в суммировании.

ПРИМЕР 7. Исследователь изучает регулярность и эффект семейных конфликтов. Он просил респондентов отметить, по каким дням прошедшей недели (переменные *MON – SUN*) споры начинались из-за денег, из-за детей или из-за родственников (соответственно коды 1, 2, 3 там). А также просил оценить по шкале от 1 до 4 степень психологического дискомфорта, который вызывает у респондента спор на эти три темы (переменные *MONEY, KIDS, RELAT*). Исследователь захотел потом узнать, сколько должно минимум проходить дней от спора до спора на одну и ту же тему, чтобы респондент оценивал свой дискомфорт такой-то степенью.

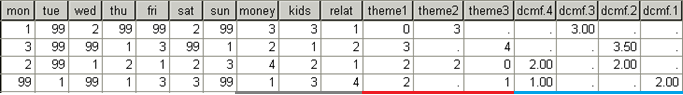
!KO\_valvars vars= mon tue wed thu fri sat sun /values= 1 2 3 /newvars= theme /newvals= ORDINAL

/redupl= MINGAP.

missing values theme1 theme2 theme3 (0).

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

/redupl= MEAN.



Первая макрос-команда заказала создать переменные *THEME1 – THEME3*, в соответствие кодам тем 1, 2, 3, содержащимся в переменных *MON – SUN*. Она затребовала, чтобы значениями в новых переменных был минимальный разрыв (redupl=MINGAP) в днях (NEWVALS=ORDINAL, день недели по порядку). Переменные *THEME1, THEME2, THEME3* показывают, «сколько минимум проходит дней от спора до спора» на тему, соответственно, денег, детей, родственников. Так, у первого респондента спор на 1-ю тему случился только однажды, поэтому разрыв=0, а на 2-ю тему спор был дважды с интервалом в 3 дня (разрыв=3).

Т.к. исследователя не интересуют однократные споры в течение недели, он придал статус пропущенного значения нулю в созданных переменных, командой MISSING VALUES.

Вторая макрос-команда заказала создать из набора *MONEY, KIDS, RELAT* с баллами дискомфорта переменные *DCMF.4 – DCMF.1*, соответствующие этим 4-м степеням дискомфорта. Эти переменные исследователь и ставил целью получить для дальнейшего анализа. Значения их, отражающие «количество дней от спора до спора», почерпнуты из переменных *THEME1 – THEME3*. Например, значения в *DCMF.4* – это сколько проходит дней от спора до спора на одну и ту же тему, которую респонденты считают дискомфортной в 4 балла. На тот случай, если в *MONEY, KIDS, RELAT* одна и та же степень дискомфорта встречается более раза, заказано усреднить (redupl=MEAN) соответствующие значения, взятые из *THEME1 – THEME3*. Так, у 2-го респондента 2-мя баллами оценен дискомфорт и 1-й темы, и 3-й темы, поэтому значения 3 и 4 (интервалы, в днях) усреднены в 3.5.

***Особые режимы***

Макрос не реагирует на взвешенность, фильтрованность (FILTER, USE), расщепленность массива данных. Он не слушается временных (под TEMPORARY) преобразований.

***Родственные макросы***

!KO\_SRSREF (см. ниже) c PACK=IREFER эквивалентен данному макросу, когда NEWVALS это переменные. Макрос !KO\_DERAND (коллекция “Derandomize tasks”) способен дать тот же эффект.

Подсчитать горизонтально частоты вы можете также макросами !KO\_HFREQ и !KO\_HCOUNT (коллекция “Horizontal tools”).

# МАКРОС !KO\_SRSREF: РАЗВЕРТКА СЕРИИ ПУНКТОВ ИЗ УПЛОТНЕННОЙ НАБИВКИ С ПОМОЩЬЮ СПРАВОЧНОГО НАБОРА ПЕРЕМЕННЫХ

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

!KO\_srsref series= *s1 to s5* /\*Переменные "серия", содержат копируемые данные;

/\*должны быть рядом, писать надо ч-з to

/refer= *c1 c2 c3 c4* /\*Переменные "справка" (MRC-набор), содержат коды выбранных пунктов;

/\*писать либо поименно, либо ч-з to

/rated= *1 to 8* EXCEPT *3 5* /\*Список кодов (неотриц числа, или поименно, или диапазоном ч-з to)

/\*всех оценивавшихся пунктов; после диапазона можно перечислить, после сл EXCEPT,

/\*коды, к-рые требуется исключить из него

/pack= MREFER /\*Характер набивки в "серии": монотонно к списку rated (MRATED),

/\*изоморфно (IREFER) или монотонно (MREFER) к набивке в REFER

/cap= *'x.'* /\*Приставка в имена новых создаваемых переменных серии

/rtimes= *1* /\*Для pack= IREFER, MREFER: сколько раз максимум может быть выбран респ-том

/\*один и тот же код (п/у=1)

/noans= *99* /\*Опционально: значение "нет оценки" в новые переменные

/\*(пункт выбран, но его оценка в series - пропущенное значение)

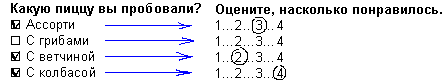
/nosel= *999* /\*Опционально: значение "пункт не выбран" в новые переменные,

/\*он станет user-missing.

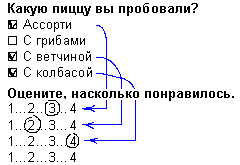
Минимум надо задать SERIES, REFER, RATED, PACK, CAP.

Вообразите вопрос, который просит как-то оценить (к примеру, по балльной шкале) пункты, которые нужно было *сначала выбрать* в другом вопросе – вопросе на множественный ответ. Респондент оценивает не все пункты списка, а только те, что выбрал прежде.

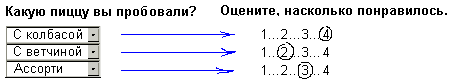
В подобной ситуации, если речь идет о ручной набивке данных, может быть неудобным набивать данные оценивания в окончательные, готовые к анализу переменные, представляющие собой все пункты списка, - потому что это потребует набивщику искать, в какую из переменных набить ответ (оценку). Или же это потребует введения программистских «правил перехода», что не всегда технически доступно. Вместо этого ответы-оценки можно бы набивать каким-л. *уплотненным* способом, что удобнее и быстрее. Иллюстрации ниже поясняют эту идею.



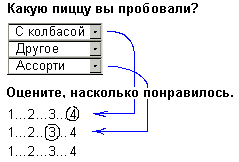
На первой схеме - не уплотненный вариант набивки. Каждому пункту, в порядке как они идут в анкете, отвечает своя переменная, куда будет набита оценка. Такие переменные уже готовы к анализу. Т.к. пункт «С грибами» не был предварительно выбран респондентом, соответствующая переменная-для-оценки будет пропущена во время набивки. Эту набивку можно назвать *изоморфной списку пунктов*: каждая переменная-для-оценки отвечает строго определенному пункту.



На этой схеме – набивка, *монотонная списку пунктов*: пункты и переменные-для-оценок соотносятся лишь по последовательности. Это уплотненная набивка, т.к. будут задействованы только *первые* три переменные-для-оценок, а 4-я останется пустой. Пункт «С грибами» не был выбран; а могло быть и так, что он хоть и был выбран, но запрещен анкетой к оцениванию, - в том и другом случае он игнорируется при набивке оценок. Таким образом, у разных респондентов будет набито разное число оценок, с разным смещением оценок в сторону первой переменной-для-оценок. Но у всех респондентов оценки должны быть набиты в общей последовательности.



Этот тип набивки – *изоморфный выборам пунктов*. В вопросе на выбор пунктов респондент мог выбирать пункты в любом порядке, *в таком* порядке и будут набиты его выбранные коды в категориальный набор множественного ответа, т.е. в порядке «С колбасой»–«С ветчиной»–«Ассорти». Переменные-для-оценок строго соответствуют набитым кодам по месту. Так, пункт «Ассорти» выбран и набит третьим; следовательно, набить оценку для «Ассорти» нужно именно в 3-й переменной-для-оценок. У разных респондентов порядок набивки пунктов и, следовательно, оценок, может быть неодинаков.



Уплотненная набивка, *монотонная выборам пунктов*, отличается от предыдущей набивки тем, что переменные-для-оценок отвечают набитым кодам лишь по последовательности. Пункты, запрещенные к оцениванию, игнорируются при набивке оценок. Так, пункт «Другое», 2-й выбранный респондентом, не был предназначен для оценивания, и во 2-ю переменную-для-оценок поэтому будет набита оценка уже следующего выбранного пункта, «Ассорти». Третья переменная-для-оценок останется пуста. Как и в предыдущем случае, у разных респондентов порядок набивки пунктов и, следовательно, оценок, может быть неодинаков.

Данный макрос предназничен для распаковки «переменных-с-оценками»[[6]](#footnote-6), набитых любым из 3-х последних способов, в набивку по первому способу, ибо только такая набивка пригодна для обработки данных. Макрос берет (i) серию уплотненно набитых оценками переменных и (ii) «справочный» набор множественного ответа, где содержатся коды выбранных респондентами пунктов, и создает из той серии новые переменные, соответсвующие всем пунктам, предназначенным анкетой к оцениванию. Эта «распакованная» SRS-серия переменных готова к анализу. Набор множественного ответа, играющий роль справки о выбранных пунктах – должен быть категориальным (MRC)[[7]](#footnote-7). Макрос пригоден и для случая, когда пункт разрешено было выбрать и оценить более одного раза.

Макрос не проверяет и не чистит данные. Он именно распаковывает данные, набитые в уплотненной манере.

***Подкоманды***

**SERIES**

Входящие числовые переменные с уплотненной («влево», т.е. к 1-й переменной) набивкой «оценок». Переменные должны быть в массиве данных рядом, писать надо через “to”: *var* to *var*. Заметьте, что уплотненная набивка не означает сжатие за счет выбранных, но неоцененных, по причине отказа оценить, пунктов. Оценка таких пунктов *должна присутствовать* во входящей серии – в виде пропуска или кода, означающего «нет ответа». Уплотнение – значит сжатие за счет пунктов не выбранных респондентом либо запрещенных к оцениванию анкетой.

В том случае, если PACK=IREFER, число переменных SERIES должно быть равно числу переменных REFER. При любой набивке оценок последовательность переменных SERIES в массиве данных должна быть без изьятия. Например, если оценки уплотненно набивались, слева направо, в переменные *VAR1 VAR2 VAR3*, вы не можете вырезать из массива или переместить переменную *VAR2* без ущерба для правильной распаковки оценок.

**REFER**

Входящие числовые категориальные переменные, содержащие (неотрицательные) коды выбранных респондентами пунктов. Играют роль справки о выбранных пунктах. Это должен быть «категориальный набор множественного ответа» (MRC), т.е. несколько категориальных переменных с общим банком вариантов ответа. Писать можно поименно или через “to”, но не смешанно. В этих данных могут присутствовать как коды (пункты), разрешенные к оцениванию, так и коды (пункты), запрещенные к оцениванию, если таковые есть. Последние макрос игнорирует.

Переменные MRC-набора не обязаны образовывать «правильное» строение, т.е. такое строение, когда левее валидного кода не позволено находиться пропуску. Набор переменных не обязан быть зарегистрирован в файле данных как MR-набор.

Единственная категориальная переменная может быть указана в качестве REFER.

**RATED**

Список пунктов, разрешенных к оцениванию - столько и будет создано SRS-переменных. Это должны быть неотрицательные коды – значения переменных REFER.

Заметьте: это список *всех* пунктов, потенциально оценивавшихся выборкой респондентов, а не отдельных пунктов, которые вас могут интересовать лишь «сейчас» для развертки в переменные-для-анализа. Макрос развертывает в SRS-серию не иначе как все пункты, которые предназначались к оцениванию анкетой и оценки которых, следовательно, хранятся в переменных-с-оценками SERIES. Если некоторый пункт был разрешен к оцениванию – так что оценки для него существуют в SERIES, – то вы не можете выпустить код этого пункта из списка RATED. Также, все коды RATED должны быть валидными значениями в переменных REFER.

Задать список можно или перечислением, или диапазоном через “to”: min to max; но не смешанно. В первом случае коды в общем не обязаны быть целыми числами. Когда же они заданы диапазоном, макрос формирует список как целые числа от min до max, которые должны быть целыми.

Вставкой ключевого слова EXCEPT после диапазона можно задать список кодов (пунктов), которые вы хотите выпустить из диапазона (например, это пункты, запрещенные к оцениванию: их не оценивали респонденты). Например, если RATED= *1 to 5* EXCEPT *2 4*, то список кодов, который сформирует макрос, будет *1 3 5*. Если после слова EXCEPT кодов нет, то само слово игнорируется.

**PACK**

Укажите тип уплотненной (т.е. сдвинутой к левому краю, к 1-й переменной) набивки в переменных SERIES.

MRATED - набивка, монотонная списку пунктов. Оценки набиты слева направо в той *последовательности*, как следуют пункты в списке RATED. Так должно быть у всех респондентов. Порядок набивки кодов в переменных REFER не играет роли.

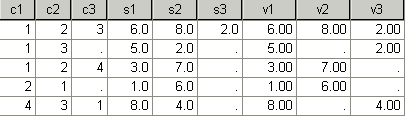
IREFER - набивка, изоморфная выборам пунктов. Оценка пункта находится в переменной SERIES *того же* порядкового номера, что переменная REFER, содержащая код этого пункта. У разных респондентов этот соответственный порядок кодов и оценок не обязан быть один и тот же. Число переменных SERIES должно быть равно числу переменных REFER.

MREFER - набивка, монотонная выборам пунктов. Оценки набиты слева направо в той *последовательности*, как набиты коды соответствующих им пунктов в переменных REFER. У разных респондентов этот соответственный порядок кодов и оценок не обязан быть один и тот же.

При набивке MRATED или MREFER, если переменных SERIES введено в макрос меньше, чем присутствует разрешенных к оцениваиню кодов (пунктов) в справочных переменных REFER, то SPSS выдаст предупреждение: “The subscript in a vector reference is missing or otherwise invalid…” с указанием номера наблюдения, в котором обнаружилась такая нехватка. Имеется в виду, что макрос искал взять оценку из очередной переменной SERIES, но очередной переменной не встретил. Указанное разногласие между числом кодов в справочном наборе REFER и числом переменных-с-оценками SERIES может быть следствием неправильно набитых данных или неправильно заданных входящих, а также следствием намеренного введения в макрос не всех существующих переменных с уплотненной набивкой оценок, а только *первых* стольких-то (последнее – допустимо).

ПРИМЕР 1. Набивка, монотонная списку пунктов.

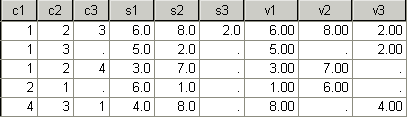
!KO\_srsref series= s1 to s3 /pack= MRATED /refer= c1 to c3 /rated= 1 2 3 /cap= v.



* «Переменные-с-оценками» *S1 – S3* содержат некоторые оценки или ответы, набитые у всех респондентов в последовательности оценивавшихся пунктов. Эти пункты (их коды) в такой последовательности указаны в подкоманде RATED.
* MRC-набор *C1 – C3* содержит коды выбранных респондентами пунктов и играет роль справки. Пункт с кодом 4 хотя и выбирался, но не предназначался потом для оценивания (в вопросе он значил «Другое»), и оценкам его не может быть места в *S1 – S3*; он не упомянут потому и в RATED.
* Макрос распаковал оценки в новые, готовые-к-анализу переменные *V1 V2 V3* (коды вошли в имена переменных).

ПРИМЕР 2. Те же данные, но набитые по способу: набивка, монотонная выборам пунктов.

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



* Набивка в переменных-с-оценками *S1 – S3* последовательностью отвечает набивке кодов в справочном наборе *C1 – C3*. (Это становится видно, если сравнить с предыдущим примером то, как набиты оценки 2-х последних респондентов.) Последовательность написания кодов в RATED не важна; она задает только очередность выходящих переменных *V1 – V3*.
* Результаты распаковки те же, что в предыдущем примере.

**CAP**

Укажите приставку в имена выходящих переменных, ее можно взять в кавычки/апострофы (они обязательны, если вместе с приставкой вы используете разделитель-точку: ‘var.’). Имена переменных будут образованы из приставки и кодов RATED. Новые переменные должны не существовать пока в массиве данных.

**RTIMES**

Эта подкоманда – для случая, если пункты, разрешенные к оцениванию, могли по условию анкетирования быть выбраны и оценены более раза. Например, вопрос на множественный ответ мог интересоваться, «Телевизоры каких производителей есть у вас в семье?», и разрешать назвать одного и того же производителя несколько раз, если в семье несколько телевизоров того же производителя. Значит, один и тот же код может повторяться в переменных REFER на правах самостоятельных пунктов, каждый из которых будет оценен в переменных SERIES своей оценкой.

Укажите здесь, сколько раз максимум мог быть выбран/оценен каждый пункт из RATED. Обычно это есть само число переменных REFER, но вы можете задать и меньшее число: в этом случае оценки только для первых RTIMES встреч кода будут перенесены в создаваемые переменные. Макрос создает для каждого пункта столько переменных, сколько задано RTIMES, и имена выходящих переменных оканчиваются на индексы, означающие «1-я встреча», «2-я встреча», и т.д.

По умолчанию/незаданию подкоманды, RTIMES=1, т.е. повторения кодов в переменных REFER не предусмотрены (повторений должно не быть). RTIMES *не действует* при PACK=MRATED: при этом типе набивки макрос не реагирует на повторения кодов в REFER, ему важно лишь, есть там код или нет.

**NOANS, NOSEL**

По умолчанию или незаданию этих подкоманд макрос оставляет в выходящих SRS-переменных пустыми ячейки, куда не было вставлено валидной оценки. Вы можете заказать, какие значения (любые числа) туда поместить.

NOANS означает «нет ответа», т.е. пункт, разрешенный к оценке, был выбран, но его оценка в переменных SERIES оказалась пропущенным значением (system- или user-missing). Значение NOANS макрос не станет делать пропуском.

NOSEL означает «пункт не выбран», т.е. кода в справочных переменных REFER не нашлось. Значение NOSEL становится user-missing в выходящих переменных. Допустимо указать одно и то же значение для NOANS и NOSEL.

В любом случае макрос оставляет пустыми (system-missing) во всей выходящей SRS-серии те наблюдения, которые были пропущены (system- или user-missing) во *всех* переменных справочного набора REFER, иными словами, это респонденты, которых не просили выбирать пункты вовсе.

ПРИМЕР 3.

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

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

* Набивка в переменных-с-оценками *I1 – I14* последовательностью отвечает набивке кодов в справочном наборе *MRC1 – MRC8*. В SERIES запакованы оценки брендов товара, закодированных в REFER кодами 1–20 (кроме бренда 19, который не предназначался для оценивания).
* Респондент мог выбрать и оценить один и тот же бренд до трех раз (RTIMES=3).
* В выходящих распакованных переменных-с-оценками (их имена начинаются на *V.*) отсутствие оценки для выбранного кода будет закодировано как 99, а если оценки нет потому, что бренд не выбран – то 999 (и этот код будет user-missing).

***Особые режимы***

Макрос не реагирует на взвешенность, фильтрованность (FILTER, USE), расщепленность массива данных. Он не слушается временных (под TEMPORARY) преобразований.

***Родственные макросы***

Если хотите «развернуть» в пункты не просто матричный вопрос, а разнородные переменные, и набивка их отвечает логике IREFER-набивки, используйте макрос !KO\_DERAND (коллекция “Derandomize tasks”).

1. Если значение встречено более раза, то только последнее зачтется. [↑](#footnote-ref-1)
2. Которые вы можете задать прямо или указать на содержащие их переменные; в последнем случае для разных наблюдений могут быть свои новые значения. [↑](#footnote-ref-2)
3. Отрицательные числа вам придется предварительно перекодировать в неотрицательные, в меню Recode. [↑](#footnote-ref-3)
4. Если такие переменные уже есть в массиве, SPSS сообщит об ошибке, однако макрос продолжит работу и изменит эти переменные. Будьте внимательны. [↑](#footnote-ref-4)
5. Вы можете регулировать зерно порождения случайных чисел в меню Random Number Generators. [↑](#footnote-ref-5)
6. В кавычках потому, что в этих переменных могут быть на самом деле всякие данные, а не обязательно шкальные оценки. Слово «оценка» в описании данного макроса употреблено условно. [↑](#footnote-ref-6)
7. Если он дихотомический (MRD), переделайте его прежде в категориальный макросом !KO\_MRDMRC (коллекция “Categorical – Binary recodings”). Только вариант с набивкой, монотонной списку пунктов, MRATED, будет возможен. [↑](#footnote-ref-7)