***Derandomize tasks***

SPSS macros by Kirill Orlov

kior@comtv.ru, ttnphns@gmail.com

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

All rights reserved

*Derandomizing of tasks.* If same tasks (some stimuli, e.g. questionnaire questions, specimens being tested, or medical treatments) were offered in different sequence to different respondents, so that the data, too, were then entered in that order of exposure – “order of trials” – then the macro will restructure these data into a unified “order of tasks” wherein each variable contains data of only one task.

**MACRO !KO\_DERAND: TAKING AWAY RANDOMIZATION OF TASKS EXPOSURE FROM DATA FILE STRUCTURE**

Version 2, May 2006 (Version 1, May 2003). Tested on SPSS 11, 11.5, 13, 14.

*This macro needs SPSS 11.0 or higher[[1]](#footnote-1).*

!KO\_derand svars= s1 s2 /\*m (m>=1) variables corresponding to trials and identifying tasks, name-by-name list

/codes= 1 2 /\*Task codes in these variables (all or needed), name-by-name list

/indxs= 1 2 /\*Indices corresponding to CODES, into the names of output variables: some numbers, characters; possible to put CODES list itself;

/\*if unspecify INDXS the output file will be single group of variables with data sewn one under another and with identificator-var CODES

/indpos= APPEND /\*Where to put indices into names: end-name (APPEND, default) or after the last non-digit (INSERT)

/separ= '.' /\*Optionally: separator (quoted) into the names of output variables; if separator not needed unspecify it or omit the subcommand

/caps= x y z /\*Prefixes into the names of output variables: either a list of p words (p – number of input variables in a group) or a single word (the program will add number from 1 to p to it)

/copy= /\*Optionally: variables (name-by-name list) simply to copy into the output file

/vars= x1 y1 z1 / x2 y2 z2 /\*m groups (in correspondence to SVARS) by p (p>=1) variables between which to shift data; separate groups by slash, write name-by-name.

Minimal specification SVARS, CODES, CAPS, VARS. VARS must go the last.

Если респондентам или испытуемым предъявлялся ряд заданий (под понятие «задания» можно подвести всякой природы альтернативные стимулы, как то: вопросы анкеты, пункты теста, дегустируемые образцы, лечебные воздействия), и порядок предъявления заданий варьировал для разных респондентов (рандомизация)[[2]](#footnote-2), то нередко и данные набиваются в этом же варьирующем от респондента к респонденту порядке – порядке «проб», или испытаний. При этом создают и особые переменные, в которых запротоколировано, какое задание было в какой пробе у данного респондента. Очередная такая переменная – это очередная проба, а значения в ней – это коды заданий.

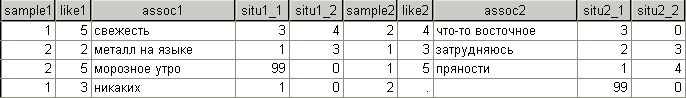
Например, если выборку просили продегустировать 2 образца и оценить их по 3-м характеристикам, то данные бывают набиты в 6 переменных: ***S1*** *X1 Y1 Z1* ***S2*** *X2 Y2 Z2*, - в этой схеме индекс в имени переменной обозначает порядковый номер предъявленной пробы; переменные **S** будут содержать в качестве значений коды образцов (образец1 или образец2), а переменные X, Y и Z это три характеристики, о которых спрашивали респондентов.

Поскольку для анализа данных такое строение обычно неудобно, требуется устранить рандомизованность. Нужно, чтобы X1, Y1, Z1 содержали только данные, касающиеся одного образца (задания), а X2, Y2, Z2 содержали только данные, касающиеся второго образца (задания). Тогда надобность в переменных **S1** и **S2** отпадет. Макрос делает как раз такое перемещение данных. Он также создает переменную, в которой сохранена информация об изначальной рандомизации заданий.

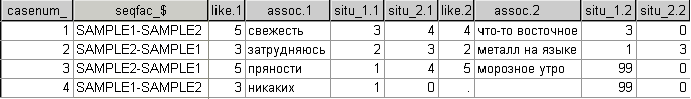
Каждому заданию может быть посвящено во входящем файле любое, но одинаковое для всех заданий число переменных с данными (ответами на задания). Эти переменные могут быть любых типов (числовые, текстовые и т.д.). Макрос создает новый рабочий файл, в котором переменные идут в порядке заданий (его указывает пользователь). Первой идет группа переменных, относящихся к одному заданию, потом такая же группа переменных, относящихся к другому заданию, и т.д. Дополнительно вы можете затребовать скопировать любые посторонние переменные из старого файла в новый.

ПРИМЕР 1. Испытуемые дегустировали две марки продукта – марку клиента и марку конкурента и отвечали на 3 вопроса: насколько вам понравился продукт; каковы ваши ассоциации к нему; в каких ситуациях вы бы употребили его (этот вопрос был на множественный ответ, до двух ответов). Марки предлагались рандомизованно.

!KO\_derand svars= sample1 sample2 /codes= 1 2 /indxs= 1 2 /caps= like assoc situ\_1 situ\_2 /separ= '.' /vars= like1 assoc1 situ1\_1 situ1\_2 / like2 assoc2 situ2\_1 situ2\_2.



Коды продуктов набиты в SAMPLE1 (1-я проба) и SAMPLE2 (2-я) и упомянуты в подкоманде CODES. Пробе 1 отвечает группа переменных LIKE1, ASSOC1 и категориальный набор множественного ответа SITU1\_1, SITU1\_2. Такая же группа переменных относится и к пробе 2. Эти 2 блока по 4 переменных указаны в подкоманде VARS через слэш.



Макрос переструктурировал данные переменных VARS так, что образцы разошлись строго по «своим» переменным. Имена этих новых переменных состоят из приставок, указанных в подкоманде CAPS с добавлением индексов, указанных в INDXS (в данном примере тождественным самим кодам CODES). Перед индексом заказан разделитель в виде точки.

Первая группа переменных (с индексом 1) это марка с кодом 1, вторая группа (индекс 2) – марка с кодом 2. CASENUM\_ это нумерация наблюдений, взятая из входящего рабочего файла в выходящий.

Переменная SEQFAC\_$ показывает, какова была изначальная рандомизация. В составленной из имен переменных SVARS цепочке первое написанное имя отвечает марке с кодом 1, а второе марке с кодом 2 – потому что именно в такой последовательности пользователь затребовал разместить марки в выходящем файле, указав CODES= 1 2. Таким образом, цепочка “SAMPLE1-SAMPLE2” говорит: код марки 1 во входящем файле располагался в переменной SAMPLE1 (что значит что марка 1 дегустировалась в первой пробе), а код марки 2 располагался в переменной SAMPLE2 (т.е. марка 2 тестировалась во второй пробе). “SAMPLE2-SAMPLE1”, соответственно, говорит о рандомизованности противоположной: «марка1 в пробе 2, а марка 2 в пробе 1». Переменную SEQFAC\_$ (или числовую переменную, какую можно получить из нее в меню Autorecode) можно использовать, – например при сравнении марок в Repeated measures ANOVA, как фактор.

***Подкоманды***

**SVARS**

Числовые переменные, одна или более, идентифицирующие задания, - т.е. куда записывались коды заданий[[3]](#footnote-3). Поименный список. Каждую переменную можно понять как очередную пробу (предъявление рандомизованного задания). Эти переменные играют роль справки об заданиях. О пропущенных значениях в них см. п/к CODES. Если переменных SVARS много, лучше если у них будут недлинные имена.

**VARS**

Эта подкоманда должна идти последней. В ней поименно укажите переменные, содержащие ответы респондентов или реакции испытуемых на задания. Если в предлагалось *m* заданий (следовательно, *m* проб) и замерялось *p* характеристик по каждому из них (*p*>=1), то вы должны указать *m* X *p* переменных. Когда характеристик несколько, одни из них могут быть числовыми, другие тестовыми, т.е. переменные могут быть всяких типов.

Имена в списке VARS должны быть сгруппированы по пробам; группы надо разделить слэшем (/). Первой идет группа, соответствующая первой переменной в списке SVARS, второй идет группа, соответствующая второй переменной SVARS, и т.д.; число групп и число переменных SVARS должно совпадать. Число переменных в группах должно быть одинаково (макрос не проверяет это, так что будьте внимательны). Переменные, одинаковые по своему порядковому номеру в группе, должны быть одного типа (числового или текстового) и, предполагается, должны обозначать одну и ту же характеристику. Поэтому их свойства (такие, как ярлык переменной, ярлыки значений, user-missing спецификации) предполагаются одинаковыми. Если по свойствам эти переменные не идентичны, в выходящий файл будут скопированы свойства, которые наблюдаются у переменных первой группы.

Входящие в макрос переменные VARS, а также SVARS, должно не содержать имен CASENUM\_, CODES\_, SVARS\_, SVARS$$$, SEQFAC\_$.

**CODES**

Поименный список кодов, идентифицирующих задания. Это значения переменных SVARS (неотрицательные числа). Не обязательно указывать все *m* заданий, вы можете указать коды лишь нужных вам заданий. Столько заданий и будет представлено в выходящем дерандомизованном файле. Порядок кодов в CODES задаст порядок заданий в нем.

*Число наблюдений в выходящем файле*. В выходящий файл попадают только те наблюдения, у которых в переменных SVARS найдены какие-л. из кодов списка CODES. Это значит, что если во входящем файле есть респонденты, которым не предлагалось *ни одно* из тех заданий, коды которых вы указали в CODES, то выходящий файл будет содержать меньше наблюдений, чем входящий. Поскольку в выходящем файле присутствует идентификатор наблюдений CASENUM\_, вы всегда сможете правильно сшить выходящий файл с изначальным, если потребуется[[4]](#footnote-4). (Вы можете также указать в подкоманде COPY переменную-идентификатор респондентов.)

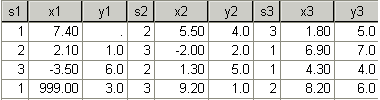
Если, по данным переменных SVARS, окажется, что респонденту предлагался одно и то же задание *более раза*, то и те и другие данные этого респондента берутся в выходящий файл – берутся как самостоятельные наблюдения, но с одним и тем же значением CASENUM\_. В этом случае выходящий файл может содержать больше наблюдений, чем входящий. Программа сообщает, что встречены такие наблюдения: "Duplicate key in a file..." и показывает их номер ("key"), т.е. значение CASENUM\_.

*Пропущенное значение* (system- или user-missing) в некоторой переменной SVARS ведет к тому, что для этого респондента в выходящий файл не будут перенесены данные из той группы переменных-характеристик, которая относится к этой SVARS-переменной. Таким образом, введением пропусков в SVARS можно ограничивать перенос данных из входящих переменных-характеристик в выходящие. При этом user-missing спецификации разных переменных SVARS макрос рассматривает независимо, что позволяет делать какие-то коды пропущенными в отдельных SVARS. Например, если хотите не брать в планируемый вами анализ данные, где задание с кодом 1 предъявлялось первым, придайте (только не под командой Temporary) user-missing статус коду 1 в той переменной SVARS, которая значит «1-я проба».

ПРИМЕР 2.

missing values s1 (1).

!KO\_derand svars= s1 s2 s3 /codes= 1 3 /indxs= 1 3 /caps= xvar yvar /separ= '.' /vars= x1 y1 /x2 y2 /x3 y3.



Переменные с кодами испытанных лекарственных препаратов (1, 2, 3), т.е. «заданий» – это S1, S2, S3, которые суть три повторных испытания. Этим трем временн**ы**м «пробам» отвечают данные трех групп переменных-характеристик, по 2 в каждой, - как и указано в подкоманде VARS (это были некие физиологические показатели в ответ на прием препарата). В подкоманде CAPS задано использовать приставки XVAR и YVAR для выходящих переменных.



Хотя препаратов три, только данные для 2-х из них, 1 и 3, судя по подкоманде CODES, заинтересовали исследователя. В результате – переменные для препарата 2 не созданы в выходящем файле.

Кроме того, исследователь затребовал игнорировать данные для случая, когда препарат 1 встречается конкретно в переменной S1: он предварительно придал статус пропущенного значения коду 1 в этой переменной командой Missing values. В результате, данные, соответствующие такому случаю, не перенесены в выходящие переменные. Значок “()” в переменной SEQFAC\_$, показывающей изначальную рандомизацию, как раз свидетельствует, что отвечающие препарату 1 данные, т.е. переменные XVAR.1 YVAR.1, отсутствуют по причине того, что валидного кода 1 не было найдено.

**INDXS**

В этой подкоманде укажите, в соответствие списку CODES, поименный список индексов[[5]](#footnote-5) в имена выходящих переменных. По индексу будет видно, к какому заданию относится та или иная выходящая переменная. Можно повторить тут сам список CODES.

Вы можете не указать список вообще (оставить пустоту после INDXS=) или умолчать подкоманду. В таком случае макрос *не станет создавать разные переменные* для разных заданий. В выходящем файле будет единственная группа переменных, и данные для заданий будут подшиты *друг под другом*; плюс будет переменная-идентификатор образцов CODES\_. Только наблюдения, для которых код образца был найден в SVARS, имеются в файле.

# INDPOS

Эта подкоманда указывает, где в имени выходящих переменных следует поместить индекс. Она служит более гибкому именованию переменных и действует, если INDXS заданы.

APPEND – ставить индекс в конце имени. Между приставкой и индексом будет разделитель SEPAR, если он задан. Тж. по умолчанию или незаданию подкоманды.

INSERT – ставить индекс сразу после последней не-цифры приставки. Разделитель SEPAR, если задан, будет стоять после индекса перед цифровым окончанием приставки. В случае когда данная приставка оканчивается не цифрой, то разделитель SEPAR после индекса опускается.

Чит. также подкоманду CAPS.

**SEPAR**

Эта необязательная подкоманда действует, если задано INDXS. В ней укажите в кавычках или апострофах знак-разделитель в имена выходящих переменных. Если он не нужен, умолчите подкоманду или опустите ее задание.

**CAPS**

Задайте приставки в имена выходящих переменных-характеристик. CAPS можно задать одним из двух способов:

*Список приставок по числу переменных в группе* – укажите столько разных приставок, сколько переменных в каждой группе переменных в VARS.

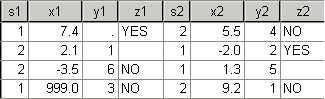
*Единственная приставка* – укажите единственную приставку. Макрос сам создаст список приставок прибавлением к ней порядкового номера переменной в группе переменных в VARS. Он сделает это, если переменных в группе более одной.

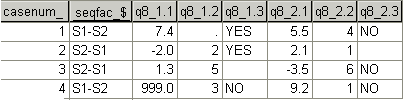
Приставки могут, по желанию, оканчиваться каким-л. знаком разделителем (например: var\_), за исключением точки.

Таким образом, имена выходящих переменных-характеристик состоят из 1) указанной вами приставки как она есть, 2) к которой макрос добавит число, если вы указали единую приставку на все переменные, 3) разделителя SEPAR, если задан, 4) индекса, помечающего данное задание. При INDPOS=INSERT индекс предшествует разделителю и встраивается между нецифровым началом и цифровым окончанием приставки (если приставка имела цифровое окончание или оно создано макросом). Помните, что в SPSS ниже 12-й версии имена переменных возможны до 8 символов.

ПРИМЕР 3. CAPS – единая приставка, INDPOS=INSERT.

!KO\_derand svars= s1 s2 /codes= 1 2 /indxs= 1 2 /caps= q8\_ /indpos= INSERT /separ= '.' /vars= x1 y1 z1 /x2 y2 z2.





В выходящем файле переменные Q8\_1.1 до Q8\_1.3 содержат данные для задания 1, а переменные Q8\_2.1 до Q8\_2.3 содержат данные для задания 2. Все они начинаются с «Q8\_» - такова заказанная единственная приставка CAPS. Макрос добавил к ней натуральные числа, так что итоговый список приставок: Q8\_1, Q8\_2, Q8\_3.

INDPOS=INSERT требует вставить индекс задания после последней не-цифры этих приставок, т.е. после «\_». Разделитель SEPAR (точка) ставится за индексом. Натуральные числа 1, 2, 3 оказываются отодвинуты в конец имени.

Если бы вместо Q8\_ в CAPS было указано Q8, то вместо, например, имени Q8\_2.3 получилось бы имя Q28.3, потому что индекс задания «2» встроился бы сразу после «Q» как последней не-цифры приставки.

**COPY**

Необязательная подкоманда, которой вы можете потребовать захватить какие-л. переменные (не упомянутые в VARS и SVARS) из входящего файла в выходящий. Эти переменные копируются без перемещения значений в них. Укажите поименный список.

***Особые режимы***

Макрос игнорирует временные (под командой TEMPORARY) преобразования перед ним, а также расщепленность файла.

***Родственные макросы***

!KO\_VALVARS и !KO\_SRSREF (“Series Response tools”) могут делать то же, не заменяя рабочий файл на новый, но требуют, чтобы все переменные-характеристики заданий были числовые и относились к единому «простому матричному вопросу».

1. SPSS 13 is known to shut down occasionally when running the macro. [↑](#footnote-ref-1)
2. В экспериментных изысканиях такой дизайн обычно называют перекрестным (crossover design). [↑](#footnote-ref-2)
3. Коды должны быть неотрицательными. Отрицательные перекодируйте заранее в неотрицательные в меню Recode. [↑](#footnote-ref-3)
4. Меню Merge files, Add variables, с пометкой Match cases on key variables… [↑](#footnote-ref-4)
5. Индексом могут служить буквы, числа, их сочетания. Числобуквы (2b) – нельзя, буквочисла (b2) – можно. [↑](#footnote-ref-5)