***Categorical - Binary recodings***

SPSS macros by Kirill Orlov

kior@akado.ru, ttnphns@gmail.com

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

All rights reserved

*Перекодировка категориальных переменных в двоичные или наоборот*. Собрание макросов для перевода категориальных данных в двоичные или обратно, например, создание фиктивных переменных или превращение наборов множественного ответа MRC (categorical multiple response) и MRD (dichotomous multiple response) – один тип в другой.

*Прочтите «*[*О SPSS макросах*](https://www.spsstools.net/ru/KO-aboutmacros)*» что они такое и как их запускать.*

*Ошибка “Protected directory”.* Некоторые из макросов, описанных в текущем документе, пишут временные файлы на жесткий диск. Если вы не обладаете полными правами Администратора вашего компьютера, это может вызвать ошибку, сообщающую среди прочего: *“SPSS Statistics cannot access a file... specifies a protected directory...”* и значащую, что дефолтная директория, какую макрос хочет использовать, защищена на вашем ПК. Чтобы решить эту проблему, в окне синтаксиса скомандуйте: CD 'myfolder'., где 'myfolder' есть путь/имя некоторой папки, куда вам разрешено сохранять файлы.

* Для создания двоичных фиктивных (dummy) переменных из категориальных переменных с единичным (альтернативным) ответом используйте [!KO\_CATBIN](#_МАКРОС_!KO_CATBIN:_ПЕРЕКОДИРОВКА), [!KO\_ACATBIN](#_МАКРОС_!KO_ACATBIN:_АВТОМАТИЧЕСКАЯ) или [!KO\_ACATBIN2](#_МАКРОС_!KO_ACATBIN2:_АВТОМАТИЧЕСКАЯ). Первый из трех для «ручной» перекодировки; два других практически одинаковы и делают «автоматическую» перекодировку.
* Для создания двоичных переменных дихотомического набора множественного ответа из переменных категориального набора множественного ответа используйте [!KO\_MRCMRD](#_МАКРОС_!MRCMRD:_ПЕРЕКОДИРОВКА_1), [!KO\_AMRCMRD](#_МАКРОС_!MRDMRC:_ПЕРЕКОДИРОВКА)  или [!KO\_AMRCMRD2](#_МАКРОС_!KO_AMRCMRD2:_АВТОМАТИЧЕСКАЯ). Первый из трех для «ручной» перекодировки; два других практически одинаковы и делают «автоматическую» перекодировку.
* Для создания переменных категориального набора множественного ответа из переменных дихотомического набора множественного ответа, либо создания категориальной переменной из фиктивных переменных, представляющих альтернативный ответ, – используйте [!KO\_MRDMRC](#_МАКРОС_!KO_MRDMRC:_ПЕРЕКОДИРОВКА). Входящие переменные не обязаны быть фактически дихотомическими.

# МАКРОС !KO\_CATBIN: ПЕРЕКОДИРОВКА ПЕРЕМЕННЫХ В ДВОИЧНЫЕ ФИКТИВНЫЕ (DUMMY)

Version 4, Apr 2013 (Version 1, Apr 1999). Tested on SPSS Statistics 20, 22, 25.

!KO\_catbin vars= *v1 v2* /\*Переменные (поименно), для которых создать двоичные (dummy) переменные

/values= *1 to 8* EXCEPT *6* /\*Список (или поименно, или диапазоном ч-з to) интересующих неотриц

/\*значений в VARS; после диапазона можно перечислить, после сл EXCEPT, значения,

/\*к-рые требуется исключить из него

/sqcond= /\*Опционально: дополнительные, связанные с послед-тью переменных ограничения для

/\*признания кода значимым ответом; одно из: GT LT GE LE EQ NE FIRST FEQ (см.)

/separ= *'\_'* /\*Опционально: разделитель в имена создаваемых двоичных переменных

/ending= /\*Окончанием имени dummy делать значение (VALUE, тж п/у) или индекс (INDEX)

/label= YES /\*Оярлычить ли созданные переменные ярлыками значений: YES или NO (тж п/у)

/macro= /\*Объединить имена dummies в наборы в виде макросов: не надо (NONE, тж п/у),

/\*все dummies (ALL) или кроме первой (BUTFIRST) или последней (BUTLAST).

Минимум надо задать VARS, VALUES.

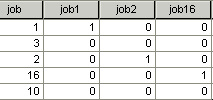
Макрос создает двоичные фиктивные (индикаторные, dummy, one-hot) переменные для указанных пользователем значений входящих переменных. Для каждого заданного значения макрос создаст из входящей переменной двоичную переменную, где 1 = наблюдение равно этому значению; 0 = наблюдение не равно этому значению. Пропуск во входящей переменной даст system-missing в образованных из нее dummy-переменных. Если входящих переменных несколько, то предложенный вами набор значений един для них всех и каждая переменная даст одно и то же число двоичных переменных.

Имена фиктивных переменных составляются из имен входящих переменных и заданных значений (или из индексов); между этими головой и хвостом может быть сепаратор. Фиктивные переменные создаются новые, поэтому таких имен должно не быть во входящем массиве.

Макрос создает временные переменные с именами, содержащими пять символов *$* подряд, например, *v$$$$$.\_2*. Поэтому желательно, чтобы таких имен не было в вашем массиве.

ПРИМЕР 1.

!KO\_catbin vars= job /values= 1 2 16.



* Переменная JOB перекодирована в три двоичные, соответствующие 3-м заданным кодам.
* Единица в двоичной переменной означает, что в JOB у текущего наблюдения именно этот код (см. имя переменной).

***Подкоманды***

**VARS**

Одна или более числовых переменных, из которых нужно создать фиктивные переменные. Поименный список. Если переменных несколько, макрос относится к ним как к самостоятельным. Обычно VARS это категориальные переменные, но не обязательно. Дробные значения допустимы в них. Ожидается, что в разных переменных хотя бы часть значений одинаковы.

**VALUES**

Список значимых кодов (значений входящих переменных), которые надо «превратить» в двоичные переменные. Это должен быть список неотрицательных чисел. Его можно задать или (i) перечислением всех нужных вам кодов (например, *1 2 5 16*), или (ii) диапазоном min to max, например, *2 to 9*. В первом случае коды не обязаны быть целыми числами. Когда же коды заданы диапазоном, то макрос формирует список как целые числа от min до max (оба эти числа должны быть целыми). Если вам важны отрицательные коды – см. ниже «задание VALUES именами переменных».

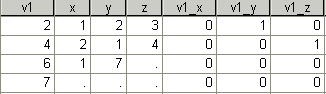
Вставкой ключевого слова EXCEPT после диапазона можно задать список кодов, которые вы хотите выпустить из диапазона. Например, если VALUES= *1 to 5* EXCEPT *2 4*, то список кодов, который сформирует макрос, будет *1 3 5*. Если после слова EXCEPT кодов нет, то само слово игнорируется.

*Задание VALUES именами переменных*. В том случае, если вы задаете VALUES перечислением, вы можете указывать в этим списке и имена числовых переменных. Таким путем можно разным респондентам указать разные значения для перекодировки, и разным респондентам задать их разное количество. Значения в этих переменных не обязаны быть неотрицательными, но должны иметь там статус валидных значений: user-missing значение будет приравнено к отсутствию значения.

Список VALUES в итоге должен составить минимум два значения.

ПРИМЕР 2. VALUES в виде списка из переменных.

!KO\_catbin vars= v1 /values= x y z /separ= '\_'.



* Требуется раскодировать в серию двоичных переменных категориальную переменную V1.
* Под признаками X, Y, Z, наличие или отсутствие которых помечают созданные переменные V1\_X, V1\_Y, V1\_Z, подразумевались в данном случае разные коды у разных респондентов. Например, у 1-го респондента признак X это код 1, но у 2-го респондента это код 2.

ПРИМЕР 3. Шкала с отрицательными значениями: -3 -2 -1 0 1 2 3.

do repeat val= -3 -2 -1 /var= m3 m2 m1.

compute var= val.

end repeat.

!KO\_catbin vars= v1 /values= m3 m2 m1 0 1 2 3 /separ= '\_'.

* Этот пример показывает, что для того, чтобы обработать отрицательные коды в V1, надо создать их в виде переменных и указать эти переменные в VALUES.

**SEPAR**

В этой необязательной подкоманде укажите символ-разделитель (или цепочку символов) в имена двоичных переменных. Разделитель зрительно удобен, когда имена входящих переменных оканчиваются цифрой. Рекомендуется взять разделитель в кавычки или апострофы, особенно если разделитель оканчивается точкой.

**ENDING**

По умолчанию и при ENDING=VALUE имя фиктивной переменной оканчивается на соответствующий ей код (значение) из п/к VALUES. Вы можете предпочесть индекс, т.е. простой порядковый номер 1, 2, 3, и т.д., в качестве окончания имени. Тогда задайте ENDING=INDEX.

**SQCOND**

Эта необязательная подкоманда действует, если входящих переменных несколько. Она дает дополнительные возможности учета порядка, в котором написаны переменные в VARS. Это может иметь смысл в ситуации, если переменные *соотносятся как «повторные измерения»* (repeated measures) чего-то одного и того же, - например, ответы респондента на сходное задание в разное время или разных условиях. Тогда вы можете захотеть положить ограничение на признание кода "значимым" (дающим единицу в соответствующей двоичной переменной), связанное с характером «динамики» ответов респондента в череде переменных. Следующие ключевые слова задают на выбор условие.

При перекодировке данной входящей переменной код у данного наблюдения может быть перекодирован в 1 только при дополнительном условии, что он:

GT - выше чем код в предыдущей входящей переменной

LT - ниже чем код в предыдущей входящей переменной

GE - выше или равен коду в предыдущей входящей переменной

LE - ниже или равен коду в предыдущей входящей переменной

EQ - равен коду в предыдущей входящей переменной

NE - не равен коду в предыдущей входящей переменной

FIRST - наблюдается у наблюдения впервые (в череде переменных VARS)

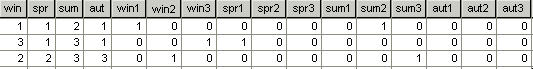
FEQ - наблюдается впервые либо равен коду в предыдущей входящей переменной.

Если условие не удовлетворяется, код перекодируется в 0, а не 1.

*Замечания*. (1) Условие не прилагается к первой переменной VARS, т.к. перед ней нет переменных. (2) Условия неравенства GT, LT, GE, LE, NE имеют в виду любой валидный код «в предыдущей переменной», т.е. значение «в предыдущей переменной» не обязано принадлежать VALUES.

ПРИМЕР 4. Исследователь собирается изучать выбор типа прохладительного напитка в сезонной перспективе, и в интересах углубления анализа хочет получить из представляющих разные сезоны категориальных переменных (коды = выбранный тип напитка) двоичные переменные. Однако его интересует не просто факт выбора данного напитка в данный сезон, а факт выбора данного напитка в данный сезон *впервые в году*.

!KO\_catbin vars= win spr sum aut /values= 1 2 3 /sqcond= FIRST.



* Четыре входящие переменные: WIN – SPR – SUM – AUT (зима – весна – лето – осень). Значения 1 2 3 это три напитка.
* Результат перекодировки WIN не зависит от SQCOND, т.к. это первая переменная VARS.
* Результат для остальных переменных: 1 только там, где данный код встречен в череде переменных VARS впервые.

**LABEL**

По умолчанию и при LABEL=NO созданные фиктивные переменные получают ярлыки переменных в виде кодов (значений) VALUES. Если задать LABEL=YES, фиктивные переменные получают ярлыки переменных в виде ярлыков этих значений, которые макрос позаимствует из свойств переменных VARS.

Если в списке VALUES есть имена переменных, исполняется LABEL=NO, даже если заказано LABEL=YES.

**MACRO**

Вы можете этой подкомандой (MACRO=ALL) объединить имена dummy-переменных, созданных из каждой данной входящей переменной, в набор, под видом макроимени. Это удобно для подстановки в синтаксис. Имя макроса составляется как !+ИмяПеременной+Separ (если SEPAR оканчивается на “.” или “\_”, то этот хвостовой символ сбрасывается). Скомандуйте MACRO=BUTFIRST (или BUTLAST), если желаете не включать имя первой (или последней) dummy-переменной в макрос (это устраняет мультиколлинерность набора, когда VALUES исчерпывают все наличные во входящей переменной валидные значения). По умолчанию, MACRO=NONE и макросы не создаются.

ПРИМЕР 5.

!KO\_catbin vars= job1 job2 /values= 1 to 13 /separ= '\_#' /macro= ALL.

frequencies !job1\_# !job2\_#.

* Макрос создал по 13 фиктивных переменных из JOB1 и JOB2 и объединил их имена в наборы под видом макросов.
* Эти имена действуют весть сеанс работы с SPSS Statistics. В данном случае они указаны в позднейшей команде FREQUENCIES, которая таким образом эквивалентна: FREQUENCIES job1\_#1 to job1\_#13 job2\_#1 to job2\_#13.

***Особые режимы***

Макрос не реагирует на взвешенность, фильтрованность (FILTER, USE), расщепленность массива данных. Он игнорирует временные (под TEMPORARY) преобразования.

***Родственные макросы***

* !KO\_HFREQ (“Horizontal tools”) также может создать dummy переменные.

# МАКРОС !KO\_ACATBIN: АВТОМАТИЧЕСКАЯ ПЕРЕКОДИРОВКА КАТЕГОРИАЛЬНЫХ ПЕРЕМЕННЫХ В ДВОИЧНЫЕ ФИКТИВНЫЕ (DUMMY)

Version 1, Dec 2013. Tested on SPSS Statistics 20, 22, 25.

!KO\_acatbin vars= *v1 v2* /\*Переменные (поименно), для которых создать двоичные (dummy) переменные

/sort= /\*Сортировать dummies по значениям в переменной (YES, тж п/у) или не надо (NO)

/separ= *'\_#'* /\*Опционально: разделитель в имена создаваемых двоичных переменных

/ending= /\*Окончанием имени dummy делать значение (VALUE, тж п/у) или индекс (INDEX)

/label= /\*Оярлычить ли созданные переменные: YES или NO (тж п/у)

/macro= /\*Объединить имена dummies в наборы в виде макросов: не надо (NONE, тж п/у),

/\*все dummies (ALL) или кроме первой (BUTFIRST) или последней (BUTLAST)

/print= YES /\*Показать частоты: YES или NO (тж п/у).

Минимум надо задать VARS.

Макрос создает двоичные фиктивные (индикаторные, dummy, one-hot) переменные из входящих целочисленных категориальных переменных. Он создает фиктивные переменные «автоматически» – для всех наличных значений входящей переменной. Для каждого наличного валидного значения макрос создаст из входящей переменной двоичную переменную, где 1 = наблюдение равно этому значению; 0 = наблюдение не равно этому значению. Пропуск во входящей переменной даст system-missing в образованных из нее dummy-переменных. Когда входящих переменных несколько, они не обязаны иметь одинаковые значения.

Макросы !KO\_ACATBIN и [!KO\_ACATBIN2](#_МАКРОС_!KO_ACATBIN2:_АВТОМАТИЧЕСКАЯ) исполняют одно и то же задание и тождественны по результату. Различаются они внутренней реализацией: !KO\_ACATBIN эксплуатирует матричную сессию, поэтому его действенность зависит от ресурса RAM-памяти; !KO\_ACATBIN2 не пользуется матричной сессией, мало зависит от RAM, он несколько быстрее, и в ситуации больших данных может быть предпочтительнее. !KO\_ACATBIN имеет подкоманду SORT, которой нет в !KO\_ACATBIN2. !KO\_ACATBIN2 имеет подкоманду DATASET, которой нет в !KO\_ACATBIN. В остальном они эквивалентны.

Имена фиктивных переменных составляются из имен входящих переменных и значений (или из индексов); между этими головой и хвостом может быть сепаратор. Фиктивные переменные создаются новые, поэтому таких имен должно не быть во входящем массиве.

Макрос создает временные переменные с именами, содержащими пять символов *$* подряд, например, *v$$$$$.\_2*. Поэтому желательно, чтобы таких имен не было в вашем массиве.

ПРИМЕР 1.

!KO\_acatbin vars= job city.

* Переменные JOB и CITY перекодируются каждая в свою серию фиктивных переменных. Фиктивных переменных в серии столько, сколько есть валидных значений во входящей переменной.

***Подкоманды***

**VARS**

Одна или более числовых переменных, из которых нужно создать фиктивные переменные. Поименный список. Если переменных несколько, макрос относится к ним как к самостоятельным. Входящие переменные должны быть категориальные. Значения (коды) в разных переменных VARS могут быть совершенно разные. Они должны быть целые (нецелые значения макрос внутренне обрезает до целых; лучше не вводите нецелые данные). Отрицательные значения разрешены (но см. п/к ENDING). В переменной должно быть минимум два неодинаковых валидных значения.

Во входящих переменных не разрешено валидное значение: *-999*, т.к. оно зарезервировано макросом.

**SORT**

По умолчанию и при SORT=YES порядок фиктивных переменных на выходе будет отвечать сортировке валидных значений в переменной (по возрастанию). При SORT=NO порядок фиктивных переменных на выходе будет отвечать очередности, в какой значения встречены в переменной (сверху вниз).

**SEPAR**

В этой необязательной подкоманде укажите символ-разделитель (или цепочку символов) в имена двоичных переменных. Разделитель зрительно удобен, когда имена входящих переменных оканчиваются цифрой. Рекомендуется взять разделитель в кавычки или апострофы, особенно если разделитель оканчивается точкой.

**ENDING**

По умолчанию и при ENDING=VALUE имя фиктивной переменной оканчивается на соответствующий ей код (значение). Вы можете предпочесть индекс, т.е. простой порядковый номер 1, 2, 3, и т.д., в качестве окончания имени. Тогда задайте ENDING=INDEX.

Если во входящей переменной есть отрицательные валидные значения, то подкоманда сработает как ENDING=INDEX для этой переменной, и макрос выдаст предупреждение, если задано было ENDING=VALUE.

**LABEL**

По умолчанию и при LABEL=NO созданные фиктивные переменные получают ярлыки переменных в виде кодов (значений). Если задать LABEL=YES, фиктивные переменные получают ярлыки переменных в виде ярлыков этих значений, которые макрос позаимствует из свойств переменных VARS.

**MACRO**

Вы можете этой подкомандой (MACRO=ALL) объединить имена dummy-переменных, созданных из каждой данной входящей переменной, в набор, под видом макроимени. Это удобно для подстановки в синтаксис. Имя макроса составляется как !+ИмяПеременной+Separ (если SEPAR оканчивается на “.” или “\_”, то этот хвостовой символ сбрасывается). Скомандуйте MACRO=BUTFIRST (или BUTLAST), если желаете не включать имя первой (или последней) dummy-переменной в макрос (это устраняет мультиколлинерность набора). По умолчанию, MACRO=NONE и макросы не создаются.

ПРИМЕР 2.

!KO\_acatbin vars= job1 job2 /separ= '\_#' /macro= BUTFIRST.

frequencies !job1\_# !job2\_#.

* Макрос создал фиктивные переменные из JOB1 и JOB2 и объединил их имена в наборы под видом макросов. Число имен в наборе на один меньше, чем число фиктивных переменных, созданных из переменной, поскольку MACRO=BUTFIRST.
* Эти имена действуют весть сеанс работы с SPSS Statistics. В данном случае они указаны в позднейшей команде FREQUENCIES.

**PRINT**

Опциональная PRINT=YES показывает в Output Viewer частоты в каждой входящей переменной. Порядок значений в показе соответствует п/к SORT.

***Особые режимы***

Макрос не реагирует на взвешенность, фильтрованность (FILTER, USE), расщепленность массива данных. Кроме того, он отменяет эти режимы в массиве. Он игнорирует временные (под TEMPORARY) преобразования.

***Родственные макросы***

* !KO\_CATCONT (“Categorical into Contrast”) создает не только двоичные dummy переменные из категориальных, но и переменные-взаимодействия между ними.
* !KO\_HFREQ (“Horizontal tools”) также может создать dummy переменные.

# МАКРОС !KO\_ACATBIN2: АВТОМАТИЧЕСКАЯ ПЕРЕКОДИРОВКА КАТЕГОРИАЛЬНЫХ ПЕРЕМЕННЫХ В ДВОИЧНЫЕ ФИКТИВНЫЕ (DUMMY)

Version 1, Oct 2021. Tested on SPSS Statistics 20, 22, 25.

*Центральная идея макроса – Рейналя Левека (Raynald Levesque)*

!KO\_acatbin2 dataset= *data* /\*Имя рабочего массива или RENAME

/vars= *v1* /\*Переменные (поименно), для которых создать двоичные (dummy) переменные

/separ= *'\_'* /\*Опционально: разделитель в имена создаваемых двоичных переменных

/ending= /\*Окончанием имени dummy делать значение (VALUE, тж п/у) или индекс (INDEX)

/label= /\*Оярлычить ли созданные переменные: YES или NO (тж п/у)

/macro= ALL /\*Объединить имена dummies в наборы в виде макросов: не надо (NONE, тж п/у),

/\*все dummies (ALL) или кроме первой (BUTFIRST) или последней (BUTLAST)

/print= YES /\*Показать частоты: YES или NO (тж п/у).

Минимум надо задать DATASET, VARS.

Макрос создает двоичные фиктивные (индикаторные, dummy, one-hot) переменные из входящих целочисленных категориальных переменных. Он создает фиктивные переменные «автоматически» – для всех наличных значений входящей переменной. Для каждого наличного валидного значения макрос создаст из входящей переменной двоичную переменную, где 1 = наблюдение равно этому значению; 0 = наблюдение не равно этому значению. Пропуск во входящей переменной даст system-missing в образованных из нее dummy-переменных. Когда входящих переменных несколько, они не обязаны иметь одинаковые значения.

Макросы [!KO\_ACATBIN](#_МАКРОС_!KO_ACATBIN:_АВТОМАТИЧЕСКАЯ) и !KO\_ACATBIN2 исполняют одно и то же задание и тождественны по результату. Различаются они внутренней реализацией: !KO\_ACATBIN эксплуатирует матричную сессию, поэтому его действенность зависит от ресурса RAM-памяти; !KO\_ACATBIN2 не пользуется матричной сессией, мало зависит от RAM, он несколько быстрее, и в ситуации больших данных может быть предпочтительнее. !KO\_ACATBIN имеет подкоманду SORT, которой нет в !KO\_ACATBIN2. !KO\_ACATBIN2 имеет подкоманду DATASET, которой нет в !KO\_ACATBIN. !KO\_ACATBIN2 разрешает значение: -999 в данных. В остальном они эквивалентны.

Имена фиктивных переменных составляются из имен входящих переменных и значений (или из индексов); между этими головой и хвостом может быть сепаратор. Фиктивные переменные создаются новые, поэтому таких имен должно не быть во входящем массиве.

Макрос создает временные переменные с именами, содержащими пять символов *$* подряд, например, *v$$$$$.\_2*. Поэтому желательно, чтобы таких имен не было в вашем массиве.

ПРИМЕР 1.

!KO\_acatbin2 dataset= DataSet0 /vars= job city.

* Переменные JOB и CITY перекодируются каждая в свою серию фиктивных переменных. Фиктивных переменных в серии столько, сколько есть валидных значений во входящей переменной.

***Подкоманды***

**DATASET**

Укажите имя массива данных, с которым работаете. Этот макрос требует указать ему массив. Или укажите DATASET=RENAME, и тогда макрос переименует входящий массив в *KO\_ACATBIN2#$.\_* и будет работать с ним под этим именем. (Необходимость п/к DATASET связана с тем, что макросу нужно знать имя рабочего массива, однако выяснять это имя для него проблематично.)

**VARS, SEPAR, ENDING, LABEL, MACRO, PRINT**

Подкоманды, идентичные таковым макроса [!KO\_ACATBIN](#_МАКРОС_!KO_ACATBIN:_АВТОМАТИЧЕСКАЯ). Как и при !KO\_ACATBIN, входящие данные должны быть целочисленные. Значение: -999 в данных разрешено.

***Особые режимы***

Макрос не реагирует на взвешенность, фильтрованность (FILTER, USE), расщепленность массива данных. Кроме того, он отменяет эти режимы в массиве. Он игнорирует временные (под TEMPORARY) преобразования.

***Родственные макросы***

* !KO\_CATCONT (“Categorical into Contrast”) создает не только двоичные dummy переменные из категориальных, но и переменные-взаимодействия между ними.
* !KO\_HFREQ (“Horizontal tools”) также может создать dummy переменные.

# МАКРОС !KO\_MRCMRD: ПЕРЕКОДИРОВКА КАТЕГОРИАЛЬНОГО НАБОРА МНОЖЕСТВЕННОГО ОТВЕТА В ДИХОТОМИЧЕСКИЙ НАБОР МНОЖЕСТВЕННОГО ОТВЕТА

Version 5, Aug 2013 (Version 1, Feb 1999). Tested on SPSS Statistics 20, 22, 25.

!KO\_mrcmrd vars= *v1 to v6* /\*Переменные, составляющие MRC, либо поименный список, либо ч-з to

/values= *1 to 8* EXCEPT *6* /\*Список (или поименно, или диапазоном ч-з to) интересующих неотриц

/\*значений в VARS; после диапазона можно перечислить, после сл EXCEPT, значения,

/\*к-рые требуется исключить из него

/\*Если указываете здесь код "нет ответа", то опустите NOANS

/dupl= /\*Как относиться к повторам значений в VARS: игнорировать повторы (NO, тж п/у);

/\*выдать результат как счетные, а не двоичные переменные (COUNT);

/\*создать свою переменную для каждого повтора (CREATE integer)

/cap= *'bin.'* /\*Приставка для имен новых создаваемых двоичных переменных (можно окавыченно)

/ending= /\*Окончанием имени переменной делать значение (VALUE, тж п/у) или индекс (INDEX)

/miss= SET /\*Делать в созданных переменных пропущенные наблюдения: (1) не надо (NO, тж п/у);

/\*(2) если все значения наблюдения в MRC-наборе missing (SET);

/\*(3) в данной переменной значение наблюдения missing (VAR имя\_переменной)

/noans= *@* /\*Окончание в имя переменной "нет ответа"; опустите, если ее не нужно создавать;

/\*после окончания можно указать ярлык

/label= YES /\*Оярлычить ли созданные MRD-переменные ярлыками значений 1-й VARS:

/\*YES или NO (тж п/у)

/regist= YES *'A Question'* /\*Регистрировать созданный MRD набор: YES или NO (тж п/у);

/\*после YES можно указать ярлык для набора.

Минимум надо задать VARS, VALUES, CAP.

**Категориальный набор множественного ответа** (Categorical multiple response set, MRC) это несколько переменных, объединенных тем, что они вместе выступают единым хранилищем данных. Данные (значения) это дискретные категории, варианты ответа на вопрос с множественным (неальтернативным) ответом; так что у одного респондента бывает набито несколько разных ответов в строке. Каждая переменная MRC-набора, таким образом, имеет смысл «еще одно отвечание респондента». Ячейки, оставшиеся незаполненными ответами, можно оставить пустыми или заполнять каким-нибудь кодом-наполнителем, которому надо придать во всех переменных набора статус пропущенного значения. Обычно стараются придать MRC-набору *правильное строение*: это такое наполнение кодами ответов, при котором каждая следующая переменная – все менее заполнена ими; другими словами, пустая/невалидная ячейка не может находиться левее валидного кода. Первая переменная, таким образом, первична. Если респондент ответов не дал, то в 1-ю переменную ему ставят код «нет ответа».

**Двоичный, или дихотомический набор множественного ответа** (Dichotomous multiple response set, MRD) это альтернативный по отношению к MRC способ хранить те же данные. В MRD-наборе переменные соответствуют категориям (вариантам ответа или признакам). Переменных столько, сколько их, а значения в переменных двоичны: 1 («да», «есть», «выбрано») и 0 («нет», «не выбрано»). Это напоминает набор фиктивных (dummy) переменных, но в отличие от фиктивных переменных MRD-набор может иметь более одной единицы в строке, поскольку MRD соответствует вопросу на множественный ответ (тогда как dummy-набор соответствует вопросу на единичный (альтернативный) ответ. Аналогично тому как в категориальных переменных обычно предусматривают код для «нет ответа на вопрос», в MRD предусматривают переменную «нет ответа на вопрос», которая равна 1, если все остальные переменные набора равны 0, и равна 0, если это не так.

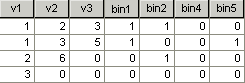
Макрос перекодирует категориальные переменные с общим пулом значений – категориальный набор множественного ответа (MRC) – в серию двоичных переменных, образующих дихотомический набор множественного ответа (MRD). Интересующие значения пользователь должен предложить. Для каждого заданного значения макрос создаст из набора входящих переменных двоичную переменную, где 1 = наблюдение равно этому значению в какой-либо из входящих переменных; 0 = наблюдение не равно этому значению ни в одной из входящих переменных. Есть опции учитывать повторения значений.

Имена двоичных переменных составляются из предложенной пользователем приставки и заданных значений (или из индексов). Переменные создаются новые, поэтому таких имен должно не быть во входящем массиве. Двоичные переменные можно объединить в MRD (зарегистрировать как набор множественного ответа).

Макрос создает временные переменные с именами, содержащими пять символов *$* подряд, например, *v$$$$$.\_2*. Поэтому желательно, чтобы таких имен не было в вашем массиве.

ПРИМЕР 1.

!KO\_mrcmrd vars= v1 v2 v3 /values= 1 2 4 5 /cap= bin.



* Переменные V1 V2 V3 указаны как MRC набор (в этом примере 0 в нем есть код-наполнитель, а остальные коды – ответы на «вопрос с множественным ответом»).
* Исследователь указал коды 1 2 4 5 для создания двоичных переменных и приставку BIN для их имен.
* Единица в двоичной переменной значит, что код присутствует в MRC-наборе.

***Подкоманды***

**VARS**

Входящие числовых переменные, составляющие для вас MRC-набор (весь или его часть); переменные не обязаны быть зарегистрированы в файле данных как MR-набор. Писать или поименно, или диапазоном через “to” (указание на диапазон смежных в массиве переменных), но не смешанно. Обычно VARS это категориальные переменные, но не обязательно. Дробные значения допустимы в них.

MRC-набор не обязан иметь *правильное строение* (см. выше); *наполнителем* может быть любой отведенный для того код (должен быть user-missing) или отсутствие кода (system-missing).

Если VARS это единственная переменная, то двоичные переменные на выходе это фиктивные (dummy) переменные.

**VALUES**

Список значимых кодов (значений входящих переменных), которые надо «превратить» в двоичные переменные. Это должен быть список неотрицательных чисел. Его можно задать или (i) перечислением всех нужных вам кодов (например, *1 2 5 16*), или (ii) диапазоном min to max, например, *2 to 9*. В первом случае коды не обязаны быть целыми числами. Когда же коды заданы диапазоном, то макрос формирует список как целые числа от min до max (оба эти числа должны быть целыми). Если вам важны отрицательные коды – см. ниже «задание VALUES именами переменных».

Вставкой ключевого слова EXCEPT после диапазона можно задать список кодов, которые вы хотите выпустить из диапазона. Например, если VALUES= *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.

Коды VALUES не обязаны быть валидными значениями в переменных VARS: !KO\_MRCMRD игнорирует user-missing статусы в VARS, когда создает двоичные переменные, отвечающие VALUES.

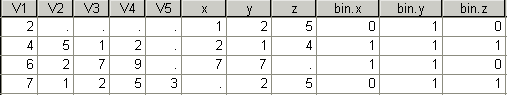
Код, значащий «нет ответа на вопрос», можно помещать в список VALUES; но лучшей альтернативой этому может быть специальная подкоманда NOANS (см.).

*Задание VALUES именами переменных*. В том случае, если вы задаете VALUES перечислением, вы можете указывать в этим списке и имена числовых переменных. Таким путем можно разным респондентам указать разные значения для перекодировки, и разным респондентам задать их разное количество. Значения в этих переменных не обязаны быть неотрицательными, но должны иметь там статус валидных значений: user-missing значение будет приравнено к отсутствию значения и даст 0 в двоичной переменной.

Список VALUES в итоге должен составить минимум два значения.

ПРИМЕР 2. VALUES как переменные.

!KO\_mrcmrd vars= v1 to v4 /values= x y z /cap= 'bin.'.



* Категориальный набор множественного ответа: V1 – V4. X, Y, Z означают три признака, и в соответствие им макрос создал три двоичные переменные с приставкой “BIN.”.
* Переменные X, Y, Z показывают, какие коды означают «есть признак X», «есть признак Y», «есть признак Z». Для разных респондентов это разные коды. Например, признак X у 1-го респондента это код 1, а у 2-го это код 2, у 3-го это код 7, а у 4-го наличие признака X невозможно (нет кода).
* У 3-го респондента код 7 означает наличие и признака X, и признака Y.

ПРИМЕР 3. Данные с отрицательным значением.

compute var= -1.

!KO\_mrcmrd vars= mrc1 to mrc4 /values= var 0 1 2 3 /cap= mrd#.

* MRC-набор содержит интересующие пользователя коды: -1, 0, 2, 3, 4. Отрицательный код -1 нельзя прямо указать в VALUES, однако можно создать из него переменную и указать ее имя в VALUES.

**DUPL**

По умолчанию/незаданию и при DUPL=NO макрос создаст одну двоичную переменную на каждый код VALUES. Единица в двоичной переменной означает, что код встречен (хотя бы раз). MRC-набор иногда содержит повторяющиеся внутри наблюдения коды (к примеру, на вопрос «Телевизоры каких производителей есть в вашей квартире?» респонденту может быть разрешено назвать несколько телевизоров одной фирмы). Если вы имеете дело с таким MRC-набором и не собираетесь игнорировать повторы, вы можете поступить одним из двух способов:

COUNT - создать переменные не двоичными, а счетными. Вместо 1 будет частота встречи кода. Подкоманда REGIST=YES будет невозможна, т.к. набор не будет дихотомическим, MRD.

CREATE *n* - создать на каждый код не одну, а *n* (целое положительное число >1) двоичных переменных. В имена двоичных переменных добавятся окончания #1, #2, и т.д., означающие «1-я встреча», «2-я встреча», и т.д. Если *n* меньше, чем реально встречается код в данных, только *n* первых его встреч в VARS будут учтены.

Опция DUPL=CREATE *n* несовместима с присутствием имен переменных в VALUES.

**CAP**

Это приставка, на которую будут начинаться имена создаваемых переменных. Вы можете взять ее в кавычки/апострофы (они рекомендуются, если оканчиваете приставку на точку: ‘bin2.’).

**ENDING**

По умолчанию и при ENDING=VALUE имя создаваемой переменной оканчивается на соответствующий ей код (значение) из п/к VALUES. Вы можете предпочесть индекс, т.е. простой порядковый номер 1, 2, 3, и т.д., в качестве окончания имени. Тогда задайте ENDING=INDEX.

**NOANS**

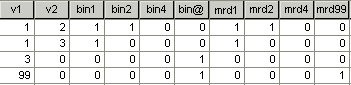
Эта подкоманда создает, в последнюю очередь, двоичную переменную со смыслом «в VARS не найдено ничего из VALUES»; и она = 1 в наблюдениях, где сумма созданных переменных, соответствующих VALUES, оказалась 0. Укажите символ (или цепочку таковых), чтобы он вошел в имя такой переменной. После символа вы можете привести, в кавычках или апострофах, ярлык для этой переменной.

Если VARS это MRC-набор, введенный полностью, и VALUES это все возможные (согласно анкете) варианты ответа на вопрос множественного ответа, тогда п/к NOANS это эквивалентная альтернатива по отношению к прямому упоминанию вами кода «нет ответа на вопрос» в VALUES.

ПРИМЕР 4. Переменная «нет ответа».

!KO\_mrcmrd vars= v1 v2 /values= 1 2 4 /cap= bin /noans= @.

!KO\_mrcmrd vars= v1 v2 /values= 1 2 4 99 /cap= mrd.



* В случае с NOANS переменная «нет ответа» BIN@ создается после остальных переменных “BIN” как негатив этих созданных переменных. Она помечает наблюдения, у которых не нашлось кодов VALUES.
* В случае, когда код 99 (в MRC-наборе он означал для исследователя «нет ответа на вопрос») подключен к VALUES, он ищется макросом во входящих переменных как обычный код, дающий свою переменную, MRD99. Наблюдение 3 с единственным кодом 3 (которого нет среди VALUES) дало во всех “MRD”-переменных 0, т.к. п/к NOANS не было.

**MISS**

Подкоманда для учета/создания пропущенных значений. По умолчанию/незаданию и при MISS=NO пустых ячеек (system-missing) в выходящих переменных не образуется. MISS позволяет сделать system-missing некоторые наблюдения. Например, это могут быть респонденты, которым не задавался вопрос. MISS делает наблюдение system-missing во *всех* созданных 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 MRC-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.

**LABEL**

По умолчанию и при LABEL=NO созданные двоичные переменные получают ярлыки переменных в виде кодов (значений) VALUES. Если задать LABEL=YES, переменные получают ярлыки переменных в виде ярлыков этих значений, которые макрос позаимствует из свойств *первой* переменной VARS.

Если в списке VALUES есть имена переменных, исполняется LABEL=NO, даже если заказано LABEL=YES.

**REGIST**

Необязательная подкоманда, регистрирующая MRD набор в .SAV-файле. Чтобы система знала, что переменные составляют двоичный набор множественного ответа. NO (тж. по умолчанию) – не регистрировать. YES – зарегистрировать; после YES можно указать в кавычках/апострофах описательный ярлык для набора, например: ‘Какие из перечисленных телепередач вы смотрите?’. Набор будет зарегистрирован под именем, указанным в CAP, из которого будет выпущен символ “\_” или “.”, если CAP на него оканчивается.

***Особые режимы***

Макрос не реагирует на взвешенность, фильтрованность (FILTER, USE), расщепленность массива данных. Он игнорирует временные (под TEMPORARY) преобразования.

***Родственные макросы***

* Если переменные вашего MRC набора представляют собой не просто попытки ответа, а ранги (1-я переменная = «1-е место», 2-я = «2-е место»…), вы можете перекодировать набор в серию переменных, значениями которых являются не 1 или 0, а эти ранги, макросом !KO\_RANKREV (“Series Response tools”).
* !KO\_HFREQ (“Horizontal tools”) также может создать частотные переменные, такие как создает DUPL=COUNT.

# МАКРОС !KO\_AMRCMRD: АВТОМАТИЧЕСКАЯ ПЕРЕКОДИРОВКА КАТЕГОРИАЛЬНОГО НАБОРА МНОЖЕСТВЕННОГО ОТВЕТА В ДИХОТОМИЧЕСКИЙ НАБОР МНОЖЕСТВЕННОГО ОТВЕТА

Version 1, Dec 2013. Tested on SPSS Statistics 20, 22, 25.

!KO\_amrcmrd vars= *v1 to v6* /\*Переменные, составляющие MRC, либо поименный список, либо ч-з to

/sort= /\*Сортировать mrd-переменные по значениям (YES, тж п/у) или не надо (NO)

/dupl= /\*Как относиться к повторам значений в VARS: игнорировать повторы (NO, тж п/у);

/\*или выдать результат как счетные, а не двоичные переменные (COUNT)

/cap= *'bin.'* /\*Приставка для имен новых создаваемых двоичных переменных (можно окавыченно)

/ending= /\*Окончанием имени двоич переменной делать значение (VALUE, тж п/у) или индекс (INDEX)

/miss= SET /\*Делать в созданных переменных пропущенные наблюдения: (1) не надо (NO, тж п/у);

/\*(2) если все значения наблюдения в MRC-наборе missing (SET);

/\*(3) в данной переменной значение наблюдения missing (VAR имя\_переменной)

/noans= *@* /\*Окончание в имя переменной "нет ответа"; опустите, если ее не нужно создавать;

/\*после окончания можно указать ярлык

/label= YES /\*Оярлычить ли созданные MRD-переменные ярлыками значений 1-й VARS:

/\*YES или NO (тж п/у)

/regist= YES *'A Question'* /\*Регистрировать созданный MRD набор: YES или NO (тж п/у);

/\*после YES можно указать ярлык для набора

/print= YES /\*Показать частоты: YES или NO (тж п/у).

Минимум надо задать VARS, CAP.

См. определение, что такое MRC и MRD, в начале описания [!KO\_MRCMRD](#_МАКРОС__!MRCMRD:).

Макрос перекодирует целочисленные категориальные переменные с общим пулом значений – категориальный набор множественного ответа (MRC) – в серию двоичных переменных, образующих дихотомический набор множественного ответа (MRD). Он создает двоичные переменные «автоматически» – для всех наличных значений совокупности входящих переменных. Для каждого наличного валидного значения макрос создаст из набора входящих переменных двоичную переменную, где 1 = наблюдение равно этому значению в какой-либо из входящих переменных; 0 = наблюдение не равно этому значению ни в одной из входящих переменных. Есть опция учитывать повторения значений (тогда переменные создаются счетными).

Макросы !KO\_AMRCMRD и [!KO\_AMRCMRD2](#_МАКРОС_!KO_AMRCMRD2:_АВТОМАТИЧЕСКАЯ) исполняют одно и то же задание и тождественны по результату. Различаются они внутренней реализацией: !KO\_AMRCMRD эксплуатирует матричную сессию, поэтому его действенность зависит от ресурса RAM-памяти; !KO\_AMRCMRD2 не пользуется матричной сессией, мало зависит от RAM. !KO\_AMRCMRD имеет подкоманду SORT, которой нет в !KO\_AMRCMRD2. !KO\_AMRCMRD2 имеет подкоманду DATASET, которой нет в !KO\_AMRCMRD. !KO\_AMRCMRD2 разрешает значение: -999 в данных. В остальном они эквивалентны.

Имена двоичных переменных составляются из предложенной пользователем приставки и заданных значений (или из индексов). Переменные создаются новые, поэтому таких имен должно не быть во входящем массиве. Двоичные переменные можно объединить в MRD (зарегистрировать как набор множественного ответа).

Макрос создает временные переменные с именами, содержащими пять символов *$* подряд, например, *v$$$$$.\_2*. Поэтому желательно, чтобы таких имен не было в вашем массиве.

ПРИМЕР 1.

!KO\_amrcmrd vars= v1 v2 v3 /cap= bin /regist= YES 'What TV channels do you watch?'.

* Переменные V1 V2 V3 указаны как MRC набор.
* Макрос создает двоичные переменные с приставкой “BIN”; их число будет равно количеству различных целых значений в V1-V3.
* REGIST=YES регистрирует двоичные переменные как MRD-набор.

***Подкоманды***

**VARS**

Входящие числовых переменные, составляющие для вас MRC-набор (весь или его часть). Писать или поименно, или диапазоном через “to” (указание на диапазон смежных в массиве переменных), но не смешанно. Входящие переменные должны быть категориальные. Их значения (коды категорий) должны быть целые (нецелые значения макрос внутренне обрезает до целых без предостережения; следует избегать ввода нецелых данных вообще). Отрицательные значения разрешены (но см. п/к ENDING). В данных должно быть минимум два неодинаковых валидных значения.

Во входящих переменных не разрешено валидное значение: *-999*, т.к. оно зарезервировано макросом.

Макрос не требует одинаковых user-missing-дефиниций во входящих переменных. Категория считается как валидная столько раз, сколько она наблюдается в данных, содержажихся в переменных, как валидная.

MRC-набор не обязан иметь *правильное строение* (см. начало описания [!KO\_MRCMRD](#_МАКРОС__!MRCMRD:)); *наполнителем* может быть любой отведенный для того код (должен быть user-missing) или отсутствие кода (system-missing).

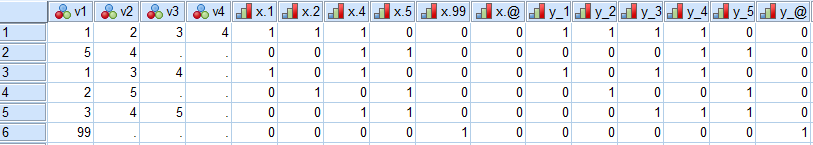
Если VARS это единственная переменная, то двоичные переменные на выходе это фиктивные (dummy) переменные.

ПРИМЕР 2. Демонстрация разного отношения !KO\_AMRCMRD и !KO\_MRCMRD к значениям входящих переменных.

missing values v1 to v4 (3).

!KO\_amrcmrd vars= v1 to v4 /cap= 'x.' /noans= @.

!KO\_mrcmrd vars= v1 to v4 /values= 1 to 5 /cap= 'y\_' /noans= @.



* Категориальный набор множественного ответа: V1 – V4. Эти переменные содержат валидные коды 1, 2, 3, 4, 5 и 99 (99 означало «нет ответа»). Команда MISSING VALUES придала коду 3 статус пропуска в V1-V4.
* !KO\_AMRCMRD перекодировал входящие переменные в двоичные X.1, X.2, X.4, X.5, X.99. Заметьте, что X.3 не создана, поскольку 3 это не валидное значение во всех V1-V4.
* !KO\_AMRCMRD также создал в конце затребованную переменную «нет ответа» X.@, но она вышла пуста – поскольку уже существует переменная X.99, созданная под валидный код 99 (означавший «нет ответа»).
* !KO\_MRCMRD перекодировал входящие переменные в двоичные Y\_1, Y\_2, Y\_3, Y\_4, Y\_5 – в соответствие заданным в VALUES кодам 1, 2, 3, 4, 5. !KO\_MRCMRD игнорирует user-missing статус заданных в VALUES кодов, и поэтому переменная Y\_3 не только создана, но и содержит единицы, как будто 3 это валидное значение в V1-V4.
* !KO\_MRCMRD создал в конце затребованную переменную «нет ответа» Y\_@. Она равна 1 там, где Y\_1–Y\_5 все равны 0. Код «нет ответа» 99 не был заказан в VALUES, и таким образом Y\_@ вышла отвечающей ему.

**SORT**

По умолчанию и при SORT=YES порядок двоичных переменных на выходе будет отвечать сортировке валидных значений во входящем MRC-наборе (по возрастанию). При SORT=NO порядок двоичных переменных на выходе будет отвечать очередности, в какой значения встречены в наборе (сверху вниз и начиная с 1-й переменной).

**DUPL**

По умолчанию/незаданию и при DUPL=NO макрос создаст одну двоичную переменную на валидный код в VARS. Единица в двоичной переменной означает, что код встречен (хотя бы раз). MRC-набор иногда содержит повторяющиеся внутри наблюдения коды (к примеру, на вопрос «Телевизоры каких производителей есть в вашей квартире?» респонденту может быть разрешено назвать несколько телевизоров одной фирмы). Если вы имеете дело с таким MRC-набором и не собираетесь игнорировать повторы, вы можете указать DUPL=COUNT. Тогда переменные создадутся не двоичными, а счетными. Вместо 1 будет частота встречи кода. Подкоманда REGIST=YES будет невозможна, т.к. набор не будет дихотомическим, MRD.

**CAP**

Это приставка, на которую будут начинаться имена создаваемых переменных. Вы можете взять ее в кавычки/апострофы (они рекомендуются, если оканчиваете приставку на точку: ‘bin2.’).

**ENDING**

По умолчанию и при ENDING=VALUE имя создаваемой переменной оканчивается на соответствующий ей код (значение). Вы можете предпочесть индекс, т.е. простой порядковый номер 1, 2, 3, и т.д., в качестве окончания имени. Тогда задайте ENDING=INDEX.

Если во входящих переменных есть отрицательные валидные значения, то подкоманда сработает как ENDING=INDEX и макрос выдаст предупреждение, если задано было ENDING=VALUE.

**NOANS**

Эта подкоманда создает, в последнюю очередь, двоичную переменную со смыслом «в VARS не найдено ни одного валидного кода»; и она = 1 в наблюдениях, где сумма созданных переменных, соответствующих валидным кодам, оказалась 0. Укажите символ (или цепочку таковых), чтобы он вошел в имя такой переменной. После символа вы можете привести, в кавычках или апострофах, ярлык для этой переменной. При ENDING=VALUE старайтесь не указывать в качестве символа голые цифры, чтобы не случилось конфликта имен переменных, что вызовет ошибку.

Если в ваших переменных VARS присутствует код «нет ответа» и он валидный, то подкоманду NOANS обычно нет смысла задавать. Также, при MISS=SET созданная п/к NOANS переменная не будет содержать единиц: все такие наблюдения заместятся system-missing.

**MISS**

Подкоманда для учета/создания пропущенных значений. По умолчанию/незаданию и при MISS=NO пустых ячеек (system-missing) в выходящих переменных не образуется. MISS позволяет сделать system-missing некоторые наблюдения. Например, это могут быть респонденты, которым не задавался вопрос. MISS делает наблюдение system-missing во *всех* созданных 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 MRC-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.

**LABEL**

По умолчанию и при LABEL=NO созданные двоичные переменные получают ярлыки переменных в виде кодов (значений). Если задать LABEL=YES, двоичные переменные получают ярлыки переменных в виде ярлыков этих значений, которые макрос позаимствует из свойств *первой* переменной VARS.

**REGIST**

Необязательная подкоманда, регистрирующая MRD набор в .SAV-файле. Чтобы система знала, что переменные составляют двоичный набор множественного ответа. NO (тж. по умолчанию) – не регистрировать. YES – зарегистрировать; после YES можно указать в кавычках/апострофах описательный ярлык для набора, например: ‘Какие из перечисленных телепередач вы смотрите?’. Набор будет зарегистрирован под именем, указанным в CAP, из которого будет выпущен символ “\_” или “.”, если CAP на него оканчивается.

REGIST=YES невозможно при DUPL=COUNT, т.к. переменные тогда могут быть не дихотомические.

**PRINT**

Опциональная PRINT=YES показывает в Output Viewer частоты во входящем MRC-наборе. Порядок значений в показе соответствует п/к SORT. Если DUPL=COUNT, то показываются частоты ответов; в ином случае показываются частоты ответивших.

***Особые режимы***

Макрос не реагирует на взвешенность, фильтрованность (FILTER, USE), расщепленность массива данных. Кроме того, он отменяет эти режимы в массиве. Он игнорирует временные (под TEMPORARY) преобразования.

***Родственные макросы***

* !KO\_HFREQ (“Horizontal tools”) также может создать частотные переменные, такие как создает DUPL=COUNT.

# МАКРОС !KO\_AMRCMRD2: АВТОМАТИЧЕСКАЯ ПЕРЕКОДИРОВКА КАТЕГОРИАЛЬНОГО НАБОРА МНОЖЕСТВЕННОГО ОТВЕТА В ДИХОТОМИЧЕСКИЙ НАБОР МНОЖЕСТВЕННОГО ОТВЕТА

Version 1, Oct 2021. Tested on SPSS Statistics 20, 22, 25.

!KO\_amrcmrd2 dataset= *data* /\*Имя рабочего массива или RENAME

/vars= *v1 to v6* /\*Переменные, составляющие MRC, либо поименный список, либо ч-з to

/dupl= /\*Как относиться к повторам значений в VARS: игнорировать повторы (NO, тж п/у);

/\*или выдать результат как счетные, а не двоичные переменные (COUNT)

/cap= *'bin.'* /\*Приставка для имен новых создаваемых двоичных переменных (можно окавыченно)

/ending= /\*Окончанием имени двоич переменной делать значение (VALUE, тж п/у)

/\*или индекс (INDEX)

/miss= SET /\*Делать в созданных переменных пропущенные наблюдения: (1) не надо (NO, тж п/у);

/\*(2) если все значения наблюдения в MRC-наборе missing (SET);

/\*(3) в данной переменной значение наблюдения missing (VAR имя\_переменной)

/noans= *@* /\*Окончание в имя переменной "нет ответа"; опустите, если ее не нужно создавать;

/\*после окончания можно указать ярлык

/label= YES /\*Оярлычить ли созданные MRD-переменные ярлыками значений 1-й VARS:

/\*YES или NO (тж п/у)

/regist= YES *'A Question'* /\*Регистрировать созданный MRD набор: YES или NO (тж п/у);

/\*после YES можно указать ярлык для набора

/print= YES /\*Показать частоты: YES или NO (тж п/у).

Минимум надо задать DATASET, VARS, CAP.

См. определение, что такое MRC и MRD, в начале описания [!KO\_MRCMRD](#_МАКРОС__!MRCMRD:).

Макрос перекодирует целочисленные категориальные переменные с общим пулом значений – категориальный набор множественного ответа (MRC) – в серию двоичных переменных, образующих дихотомический набор множественного ответа (MRD). Он создает двоичные переменные «автоматически» – для всех наличных значений совокупности входящих переменных. Для каждого наличного валидного значения макрос создаст из набора входящих переменных двоичную переменную, где 1 = наблюдение равно этому значению в какой-либо из входящих переменных; 0 = наблюдение не равно этому значению ни в одной из входящих переменных. Есть опция учитывать повторения значений (тогда переменные создаются счетными).

Макросы [!KO\_](#_МАКРОС_!MRDMRC:_ПЕРЕКОДИРОВКА)[AMRCMRD](#_МАКРОС_!MRDMRC:_ПЕРЕКОДИРОВКА) и !KO\_AMRCMRD2 исполняют одно и то же задание и тождественны по результату. Различаются они внутренней реализацией: !KO\_AMRCMRD эксплуатирует матричную сессию, поэтому его действенность зависит от ресурса RAM-памяти; !KO\_AMRCMRD2 не пользуется матричной сессией, мало зависит от RAM. !KO\_AMRCMRD имеет подкоманду SORT, которой нет в !KO\_AMRCMRD2. !KO\_AMRCMRD2 имеет подкоманду DATASET, которой нет в !KO\_AMRCMRD. !KO\_AMRCMRD2 разрешает значение: -999 в данных. В остальном они эквивалентны.

Имена двоичных переменных составляются из предложенной пользователем приставки и заданных значений (или из индексов). Переменные создаются новые, поэтому таких имен должно не быть во входящем массиве. Двоичные переменные можно объединить в MRD (зарегистрировать как набор множественного ответа).

Макрос создает временные переменные с именами, содержащими пять символов *$* подряд, например, *v$$$$$.\_2*. Поэтому желательно, чтобы таких имен не было в вашем массиве.

ПРИМЕР 1.

!KO\_amrcmrd2 dataset= DataSet0 /vars= v1 v2 v3 /cap= bin /regist= YES 'What TV channels do you watch?'.

* Переменные V1 V2 V3 указаны как MRC набор.
* Макрос создает двоичные переменные с приставкой “BIN”; их число будет равно количеству различных целых значений в V1-V3.
* REGIST=YES регистрирует двоичные переменные как MRD-набор.

***Подкоманды***

**DATASET**

Укажите имя массива данных, с которым работаете. Этот макрос требует указать ему массив. Или укажите DATASET=RENAME, и тогда макрос переименует входящий массив в *KO\_AMRCMRD2#$.\_* и будет работать с ним под этим именем. (Необходимость п/к DATASET связана с тем, что макросу нужно знать имя рабочего массива, однако выяснять это имя для него проблематично.)

**VARS, DUPL, CAP, ENDING, NOANS, MISS, LABEL, REGIST, PRINT**

Подкоманды, идентичные таковым макроса [!KO\_AMRCMRD](#_МАКРОС_!MRDMRC:_ПЕРЕКОДИРОВКА). Как и при KO\_AMRCMRD, входящие данные должны быть целочисленные. Значение: -999 в данных разрешено.

***Особые режимы***

Макрос не реагирует на взвешенность, фильтрованность (FILTER, USE), расщепленность массива данных. Кроме того, он отменяет эти режимы в массиве. Он игнорирует временные (под TEMPORARY) преобразования.

***Родственные макросы***

* !KO\_HFREQ (“Horizontal tools”) также может создать частотные переменные, такие как создает DUPL=COUNT.

# МАКРОС !KO\_MRDMRC: ПЕРЕКОДИРОВКА ДИХОТОМИЧЕСКОГО НАБОРА МНОЖЕСТВЕННОГО ОТВЕТА В КАТЕГОРИАЛЬНЫЙ НАБОР МНОЖЕСТВЕННОГО ОТВЕТА

Version 4, Sep 2014 (Version 1, Oct 2000). Tested on SPSS Statistics 20, 22, 25.

*Этот макрос требует SPSS Statistics 17 или выше*.

!KO\_mrdmrc vars= *v1 v2 v3 v4* /\*Поименно дихотомические переменные

/selval= *1* /\*Значение(я) "выбрано" в них (п/у=1): неотриц число(а) списком или

/\*диапазоном ч-з "THRU"; либо COUNT n (см.)

/codes= /\*Коды для создаваемого MRC, в соответствие VARS: либо список неотр чисел числом VARS,

/\*либо EXTRACT (взять коды из имен VARS), либо ORDINAL (породить порядк числа, тж п/у)

/cap= *'cat.'* /\*Приставка для имен новых создаваемых категориальных переменных (можно окавычить)

/limit= /\*Сколько переменных создать: необходимый минимум (MIN, п/у), столько же сколько

/\*входящих переменных (MAX), или сколько указано (число)

/filler= *0* /\*Опционально: код-заполнитель для набора

/noans= *99 'NA'* /\*Опционально: код для "нет ответа" (после него можно ярлык)

/miss= SET /\*Делать в созданных переменных пропущенные наблюдения: (1) не надо (NO, тж п/у);

/\*(2) если все значения наблюдения в MRD-наборе missing (SET);

/\*(3) в данной переменной значение наблюдения missing (VAR имя\_переменной)

/label= YES /\*Оярлычить ли созданные категории ярлыками MRD-переменных: YES или NO (тж п/у)

/regist= YES *'A Question'* /\*Регистрировать созданный MRC-набор: YES или NO (тж п/у);

/\*после YES можно указать ярлык для набора

/dropind= /\*Сбросить (YES) индекс 1 из имени, если переменная вышла единственная;

/\*п/у = NO.

Минимум надо задать VARS, CAP.

См. определение, что такое MRC и MRD, в начале описания [!KO\_MRCMRD](#_МАКРОС__!MRCMRD:).

Макрос перекодирует набор двоичных переменных или любых других числовых переменных, передающих статус двоичного признака (есть признак или же нет, либо сколько раз есть), в категориальные переменные. Если входящий набор переменных представлял собой неальтернативные признаки, т.е. был «дихотомическим набором множественного ответа» (MRD), то из него получится категориальный набор множественного ответа. Если же входящий набор представлял собой альтернативные признаки, т.е. dummy-переменные, то из него получится единственная категориальная переменная.

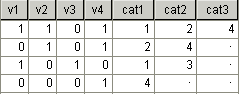
Каждой входящей переменной, т.е. каждому признаку или варианту ответа, отвечает в выходящих переменных свой числовой код. Эти коды вы можете предложить или попросить макрос сформировать автоматически. Макрос сначала создает категориальные переменные пустыми, а затем заполняет их кодами слева направо в порядке признаков (т.е. входящих переменных, как они указаны в VARS). MRC-набор на выходе имеет *правильное строение* (что это – см. в начале описания [!KO\_MRCMRD](#_МАКРОС__!MRCMRD:)).

Имена категориальных переменных составляются из предложенной пользователем приставки и порядковых индексов 1, 2, ... Переменные создаются новые, поэтому таких имен должно не быть во входящем массиве. Категориальные переменные можно объединить в MRC (зарегистрировать как набор множественного ответа).

Макрос создает временные переменные с именами, содержащими пять символов *$* подряд, например, *v$$$$$.\_2*. Поэтому желательно, чтобы таких имен не было в вашем массиве.

ПРИМЕР 1.

!KO\_mrdmrc vars= v1 v2 v3 v4 /cap= 'cat' /regist= YES 'An MRC set'.



* 4 двоичные переменные V1 V2 V3 V4 соответствуют, по умолчанию подкоманды CODES, кодам 1 2 3 4.
* Созданы категориальные переменные с приставкой “CAT”, заполненные кодами слева направо. Код вставляется, если в данной входящей переменной признак «есть» (1).
* Эти переменные зарегистрированы в файле как MRC-набор.
* Если бы V1 V2 V3 V4 были фиктивными переменными, т.е. они содержали бы вместе не более одной единицы в строке, то макрос создал бы, по умолчанию, единственную категориальную переменную CAT1.

***Подкоманды***

**VARS**

Поименный список входящих переменных (минимум две): MRD-набор множественного ответа или любые переменные, которые исследователь принимает за набор признаков (каждая переменная – один признак, который или «есть», или его «нет»). Переменные не обязаны быть фактически двоичными (1 и 0) и вообще дихотомическими или дискретными. Подкоманда SELVAL позволяет указать код или множество значений, означающие «признак есть». Пропуск в переменной приравнивается к «признака нет», т.е. к отсутствию кода «признак есть».

**SELVAL**

Эта подкоманда информирует макрос, какое значение или значения в переменных VARS (в них всех) надо понимать как статус «признак есть», или «выбрано» (а всякое иное значение, как и пропуск, будет считаться «признака нет», или «не выбрано»). По умолчанию, SELVAL=1, т.е. макрос будет считать значение 1 за «есть», а всякое иное значение за «нет». SELVAL также используется для информирования макроса, когда VARS – частотные переменные, а не статусные.

1) **Когда VARS это переменные статуса** (показывают, что признак есть или его нет). Перечислите в SELVAL значения, соответствующие статусу «признак есть». Это должны быть неотрицательные значения (десятичные дробные – допустимо). Например: SELVAL= 6 8 9. Тут любое из трех указанных значений будет пониматься как «признак есть». Повторение значения, например, SELVAL= 6 6 8 9, не вызовет ошибки. Если заданное значение не валидно в некоторой переменной VARS, оно не считается в ней как «признак есть».

Если в VARS целый диапазон значений отвечает статусу «признак есть», вы можете указать его с помощью кл. слова THRU. Например: SELVAL= 1 THRU 5, где второе указанное число должно быть не меньше, чем первое. Всякое валидное значение от 1 до 5, включительно, будет означать «признак есть». Вы можете задать SELVAR или перечислением, или диапазоном, но не смешанно.

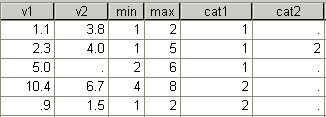
*Имена переменных в SELVAL*. На месте всех или некоторых чисел можно указывать имена переменных, которые содержали бы значения, означающие «признак есть». Это значит, что вы имеете возможность задать разные такие значения для разных респондентов (наблюдений массива). Например: SELVAL=*SELCODE* отсылает к некоторой переменной *SELCODE*, значения которой это значения, означающие «признак есть» в VARS. Значения в подобных переменных не обязаны быть неотрицательными. Если вы задаете SELVAL диапазоном, *X* THRU *Y*, то обе переменные *X* и *Y* должны быть наполнены валидными значениями и второе значение должно быть не ниже первого.

2) **Когда VARS это переменные счета** (показывают, сколько раз признак есть). Значения в переменных VARS должны быть целые неотрицательные (нецелые значения макрос внутренне обрежет до целых, к отрицательным значениям он относится как к 0). Они суть частота – количество раз, сколько наблюдается данный признак (VARS-переменная). Вы должны задать подкоманду в таком виде: SELVAL= COUNT *n*, где *n* это целое положительное число. Пусть значение VARS-переменной *X* у наблюдения *i* есть *f*. Если *f*>0 (и валидно в *X*), то макрос создаст min(*f*, *n*) копий кода, соответствующего *X*, у *i* в выходящих переменных.

Таким образом, SELVAL= COUNT *n* позволяет создать из счетных входящих переменных MRC-набор с повторяющимися категориями (ответами).

ПРИМЕР 2. Континуальные входящие переменные. И SELVAL как переменные.

!KO\_mrdmrc vars= v1 v2 /selval= min THRU max /cap= cat.



* Переменные V1 и V2 – континуальные признаки. Если значение лежит в пределах между MIN и MAX (а это имена переменных в данном случае), то считается, что признак – «есть». Эти пределы – разные у разных респондентов.
* Категориальный набор множественного ответа CAT1-CAT2 получен из этих данных. Например, у 1-го респондента признак V1 (код 1) есть, потому что значение 1.1 укладывается в диапазон между 1 и 2, а признака V2 (код 2) нет, т.к. 3.8 – не укладывается.

ПРИМЕР 3. SELVAL – отрицательное значение.

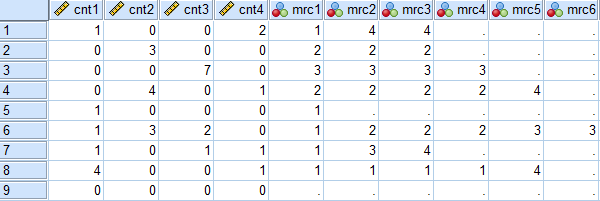
compute sel= -1.

!KO\_mrdmrc vars= v1 v2 /selval= sel /cap= cat.

* В V1 и V2 наличие признака кодировано значением: -1. В подкоманде SELVAL не разрешено прямо указать отрицательное число, но можно указать имя переменной. Последняя может иметь любые значения.

ПРИМЕР 4. Создание MRC с повторами ответов с помощью SELVAL= COUNT n.

!KO\_mrdmrc vars= cnt1 cnt2 cnt3 cnt4 /cap= mrc /selval= COUNT 4.



* Переменные CNT1 – CNT4 соответствуют 4-м медицинским жалобам. Но переменные не двоичны (есть vs нет жалобы), а счетны: сколько дней в неделю у пациента присутствует жалоба.
* Эти данные перекодированы к категориальный набор (переменные с приставкой “MRC”) с кодами 1 – 4, отвечающими 4-м жалобам.
* У респондента каждая жалоба представлена столькими копиями кода, сколько дней в неделю он испытывал жалобу; но не более 4-х копий (поскольку SELVAL= COUNT 4).

**CODES**

Каждая переменная из VARS означает признак. Макрос должен придать в соответствие признаку числовой код, которым в выходящих переменных этот признак (категория) будет кодирован. Подкоманда задает список кодов – значений будущих категориальных переменных. Есть три способа задания:

*список кодов* - перечислите коды. Это могут быть любые неотрицательные числа в количестве, сколько входящих переменных. Они соотносятся с ними по порядку.

ORDINAL - (тж. по умолчанию/незаданию) макрос формирует список автоматически как натуральные числа 1, 2, …, число\_входящих\_переменных.

EXTRACT - макрос пытается экстрагировать коды из имен входящих переменных. Это могут быть только целые числа. Входящие переменные должны все оканчиваться на эти числа-коды. Например, переменная с именем *VAR2* даст код 2. Переменная с именем *MRD3.2* также даст код 2. Макрос берет целое число из хвоста имени.

ПРИМЕР 5.

!KO\_mrdmrc vars= bin10.4 bin3.2 bin6 zxc\_11 /cap= 'mrc\_' /codes= 4 2 6 11.  
!KO\_mrdmrc vars= bin10.4 bin3.2 bin6 zxc\_11 /cap= 'mrc.' /codes= EXTRACT.

* Обе команды в этом примере эквивалентны.

**LIMIT**

Сколько создать переменных. По умолчанию и при LIMIT=MIN создается минимально-необходимое для размещения всех данных количество категориальных переменных. LIMIT=MAX создает столько переменных, сколько входящих переменных, т.е. с запасом. (При SELVAL= COUNT *n* это число еще умножается на *n*.) LIMIT=*число* создаст это число (>=1) переменных. Задав заведомо заниженное число, вы тем самым отсечете часть ответов у респондентов, а задав завышенное число, получите дополнительные пустые переменные.

**CAP**

Это приставка, на которую будут начинаться имена создаваемых категориальных переменных. Вы можете взять ее в кавычки/апострофы (они рекомендуются, если оканчиваете приставку на точку: ‘cat2.’). Имена будут состоять из приставки и порядкового индекса.

**FILLER**

Вы можете указать код-наполнитель для MRC-набора. Это может быть любое число (даже отрицательное), которого нет в CODES. Наполнителю в MRC-наборе макрос присваивает статус пропущенного значения. В том особом случае, когда код-наполнитель совпадает с заданным вами кодом «нет ответа» (NOANS), наполнитель не делается пропущенным значением в *первой* переменной набора. Это в принципе корректно потому, что MRC имеет *правильное* строение. Впрочем, не рекомендуется совмещать код-наполнитель и код «нет ответа».

**NOANS**

Этой необязательной подкомандой запрашивается создать код «нет ответа на вопрос», точнее, «нет ни одного признака из VARS». Укажите неотрицательное число. Код появится только в *первой* переменной MRC, в тех наблюдениях, которые оказались незаполненными кодами CODES.

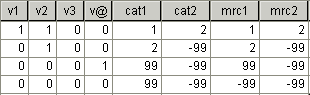
Альтернативой NOANS может быть задание среди входящих переменных VARS переменной «нет ответа», однако это эквивалентно лишь в том случае, если MRD-набор введен целиком, и данные в переменной «нет ответа» правильны. Дублирование кода «нет ответа» в списке кодов (CODES) допускается, корректность этого зависит от соображений. NOANS может быть любым неотрицательным числом. О совпадении NOANS с наполнителем см. в FILLER.

После кода вы можете указать окавыченный ярлык для него. Он будет использован при LABEL=YES.

ПРИМЕР 6. Два способа задать код «нет ответа».

!KO\_mrdmrc vars= v1 v2 v3 /codes= 1 2 3 /cap= cat /filler= -99 /noans= 99.

!KO\_mrdmrc vars= v1 v2 v3 v@ /codes= 1 2 3 99 /cap= mrc /filler= -99.



* В первом случае перекодируются переменные V1 V2 V3. Заказано вставить код «нет ответа на вопрос» (99). 3-й и 4-й респонденты не выбрали ничего в V1 V2 V3, и потому имеют 99 в CAT1.
* Во втором примере не заказано вставить код «нет ответа», а используется существующая переменная «нет ответа» V@ как еще одна входящая переменная; ей в соответствие придан код 99 в CODES. Однако данные в V@ не были вполне надежны: у 4-го респондента там должна бы быть единица. В результате в переменной MRC1 сохранился код-наполнитель (-99).

**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 и NOANS. MISS позволяет сделать system-missing некоторые наблюдения. Например, это могут быть респонденты, которым не задавался вопрос. MISS делает наблюдение system-missing во *всех* созданных 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.

**LABEL**

Вы можете этой подкомандой (LABEL=YES) оярлычить созданные категориальные значения ярлыками входящих переменных. Оярлычение значений макрос сделает только в *первой* переменной созданного категориального набора, т.к. этого достаточно для функционирования MRC-набора. По умолчанию и при LABEL=NO значения не получают ярлыков. LABEL=YES требует установленного модуля Custom Tables, поскольку внутренне использует команду CTABLES.

**REGIST**

Необязательная подкоманда, регистрирующая MRC набор (если категориальных переменных вышло больше одной) в .SAV-файле. Чтобы система знала, что переменные составляют категориальный набор множественного ответа. NO (тж. по умолчанию) – не регистрировать. YES – зарегистрировать; после YES можно указать в кавычках/апострофах описательный ярлык для набора, например: ‘Какие из перечисленных телепередач вы смотрите?’. Набор будет зарегистрирован под именем, указанным в CAP, из которого будет выпущен символ “\_” или “.”, если CAP на него оканчивается. Первая переменная набора будет также оярлычена, когда после YES дан ярлык.

**DROPIND**

Эта необязательная подкоманда действует, только если LIMIT=MIN (что есть по умолчанию). Когда созданная макросом переменная оказалась единственная (что ожидается, если VARS это фиктивные переменные, а не набор множественного ответа), то DROPIND=YES уберет индекс 1 из ее имени. Например, если CAP= ‘CAT’, переменная на выходе из макроса получит имя *CAT*, а не *CAT1*. Причем если CAP оканчивается на символ “\_” или “.”, он тоже будет сброшен. По умолчанию DROPIND=NO.

***Особые режимы***

Макрос не реагирует на взвешенность, фильтрованность (FILTER, USE), расщепленность массива данных. Кроме того, он отменяет взвешенность и расщепленность, если LABEL=YES. Он игнорирует временные (под TEMPORARY) преобразования.