***Horizontal tools***

SPSS macros by Kirill Orlov

kior@akado.ru, ttnphns@gmail.com

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

All rights reserved

*Некоторые горизонтальные операции.* Собрание макросов, исполняющих некоторые нужные вещи (такие как стандартизация, сортировка, ранжирование, категоризация или подсчет частот и уникальных значений) внутри наблюдений, горизонтально.

*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 горизонтального перешкалирования (стандартизация и т.п.) пользуйтесь [!KO\_HRESC](#_МАКРОС_!KO_HRESC:_ПЕРЕШКАЛИРОВАНИЕ).
* Для горизонтального ранжирования пользуйтесь [!KO\_HRANK](#_МАКРОС_!HRANK:_РАНЖИРОВАНИЕ).
* Для горизонтальной сортировки или рандомизации пользуйтесь [!KO\_HSORT](#_МАКРОС_!KO_HSORT:_СОРТИРОВКА), а также [!KO\_HQSORT](#_МАКРОС_!KO_HQSORT:_БЫСТРАЯ).
* Для горизонтальной категоризации количественных данных на равночастотные категории – [!KO\_HBINEF](#_МАКРОС_!KO_HBINEF:_КАТЕГОРИЗАЦИЯ).
* Для горизонтальной категоризации количественных данных на равноинтервальные категории – [!KO\_HBINEI](#_МАКРОС_!KO_HBINEI:_КАТЕГОРИЗАЦИЯ).
* Для подсчета по горизонтали частот значений – [!KO\_HFREQ](#_МАКРОС_!KO_HFREQ:_ЧАСТОТЫ,).
* Для подсчета по горизонтали разнообразия и повторяемости значений – [!KO\_HCOUNT](#_МАКРОС_!KO_HCOUNT:_ПОДСЧЕТ).
* Для замера разрывов (разниц) между значениями в их вариационном ряду – [!KO\_HGAPS](#_МАКРОС_!KO_HGAPS:_СТАТИСТИКА).

Эти макросы, кроме !KO\_HFREQ, не используют переструктурирование или транспонирование массива данных. Они основаны на преобразовательных командах.

# МАКРОС !KO\_HRESC: ПЕРЕШКАЛИРОВАНИЕ ПО ГОРИЗОНТАЛИ

Version 2, Dec 2020 (Version 1, May 2009). Tested on SPSS Statistics 20, 22, 25.

!KO\_hresc vars= *v1 v3 to v10* /\*Числовые переменные (поименно и/или ч-з to), значения которых

/\*горизонтально перешкалировать (минимум две)

/cap= *'tr\_'* /\*Приставка (можно окавыченно) в имена выходящих переменных; если не задать,

/\*сами VARS будут преобразованы

/trans= Z /\*Преобразование: Z, C, SS, SUM, RANGE, RESCALE, EXP, CLR, ZABS,

/\*DMED, SMED1, SMED2 (см.)

/discret= /\*Опционально: дискретизировать выходящие значения: RND d или TRUNC d

/missing= /\*Брать user-missing значения в обработку (INCLUDE, возможно только при

/\*незаданной CAP) или не брать (EXCLUDE, тж п/у)

/nvalid= *7* /\*Опционально: опустошить наблюдение с числом валидных значений меньше порога

/\*(укажите положит. число)

/flag= YES /\*Если задано CAP: посчитать валидные значения и пометить наблюдения-константы:

/\*YES или NO (тж п/у).

Минимум надо задать VARS, TRANS.

Перешкалирует значения числовых переменных по горизонтали, т.е. внутри наблюдений. Под общим словом «перешкалирование» тут понимаются разного рода центрации, стандартизации и нормирования. Вы можете затребовать выдать преобразованные данные как новые переменные или же заменить ими сами входящие переменные.

В рабочем массиве должно не быть переменных *NVARS###* и *NVARS##$*.

ПРИМЕР 1.

!KO\_hresc vars= v1 to v10 /cap= z\_ /trans= Z.

* Данные переменных V1 – V10 стандартизуются по горизонтали, образуя переменные с Z-баллами Z\_1 – Z\_10.

***Подкоманды***

**VARS**

Укажите числовые переменные (минимум две), значения которых преобразовывать по горизонтали. Список поименно и/или ч-з «to». Если переменных очень много, рекомендуется использовать «to»: это может сэкономить время исполнения. Данные воспринимаются как количественные (не номинальные).

**CAP**

Приставка в имена выходящих переменных. Ее можно взять в кавычки/апострофы (они рекомендуются, если приставку вы оканчиваете на точку: “var.”). Выходящие переменные будут поименованы приставкой с индексом от 1 до число входящих переменных. Во входящем массиве должно не быть переменных с этими именами.

Если не задать или умолчать подкоманду CAP, то сами входящие переменные VARS будут выходящими: все значения в них будут заменены, а любые user-missing дефиниции отменены. Ярлыки значений останутся прежние, но станут неуместны, т.к. значения переменных будут новые.

**TRANS**

Укажите требуемое преобразование данных, которое будет сделано индивидуально внутри каждого вектора («вектор» это наблюдение массива, переменные VARS).

Z - **z-стандартизация**. Из значения вычитается средняя в векторе и результат делится на стандартное отклонение (посчитанное на “df=n-1”) в векторе. В результате в векторе средняя будет 0 и стандартное отклонение 1. Если вектор был константа, макрос сделает его нулевым вектором.

C - **центрация** (отклонение от средней). Из значения вычитается средняя в векторе. В результате в векторе средняя будет 0.

SS - **нормирование к сумме квадратов 1**. Значение делится на корень квадратный суммы квадратов в векторе. В результате в векторе сумма квадратов станет 1. Если вектор был нулевая константа, макрос оставит его как был.

SUM - **нормирование к сумме 1**. Эта опция требует неотрицательных данных. Значение делится на сумму в векторе. В результате в векторе сумма будет 1. Если вектор был нулевая константа, макрос оставит его как был.

RANGE - **приведение к диапазону 1**. Значение делится на диапазон в векторе. В результате в векторе диапазон станет 1. Если вектор был константа, макрос оставит его как был.

RESCALE - **перешкалирование в диапазон [0,1]**. Из значения вычитается минимальное значение в векторе и делится на диапазон в векторе. В результате в векторе значения будут иметь диапазон [0,1]. Если вектор был константа, макрос сделает его константой 0.5.

ZABS - **стандартизация среднеабсолютным отклонением** (average absolute deviation). Значение переводится в отклонение от средней и делится на среднюю абсолютных величин этих отклонений. В результате в векторе средняя будет 0 и среднеабсолютное отклонение от средней будет 1. Если вектор был константа, макрос сделает его нулевым вектором.

DMED - **отклонение от медианы**. Из значения вычитается медиана в векторе. В результате в векторе медиана будет 0.

SMED1 - **медианная стандартизация** **среднеабсолютным отклонением** (average absolute deviation around median). Значение переводится в отклонение от медианы и делится на среднюю абсолютных величин этих отклонений. В результате в векторе медиана будет 0 и среднеабсолютное отклонение от медианы будет 1. Если вектор был константа, макрос сделает его нулевым вектором.

SMED2 - **медианная стандартизация медианноабсолютным отклонением** (median absolute deviation around median). Значение переводится в отклонение от медианы и делится на медиану абсолютных величин этих отклонений. В результате в векторе медиана будет 0 и медианноабсолютное отклонение от медианы будет 1. Если вектор был константа, макрос сделает его нулевым вектором.

EXP - нелинейное преобразование **экспоненциальное нормирование**, или **softmax**. Это нормирование к сумме 1, сделанное после экспоненциации значений (*e*val). В результате в векторе значения будут неотрицательны, и их сумма будет 1.

CLR - нелинейное преобразование **centered logratio transform**. Эта опция требует положительных данных. Сначала значения вектора нормируются к сумме 1. В полученном векторе значения делятся на геометрическую среднюю в нем, и логарифмируются. В результате в векторе сумма станет 0.

Памятуйте, что некоторые из этих преобразований требуют неотрицательных или положительных входящих данных. Макрос не проверяет на входе, годятся ли данные. Входящие наблюдения-константы вы можете отследить подкомандой FLAG (см.).

**DISCRET**

Опциональная подкоманда для дискретизации значений после преобразования. Укажите ключевое слово RND или TRUNC, означающее метод – округление или усечение, и после него – цифру 0, 1, 2 или 3, означающую число десятичных цифр для оставления в значении.

**MISSING**

Как относиться к пользовательским пропущенным (user-missing) значениям во входящих переменных: не брать их в преобразование, приняв за system-missing (MISSING=EXCLUDE, тж по умолчанию/незаданию) или взять, как валидные значения (MISSING=INCLUDE, разрешено только при незаданной п/к CAP).

**NVALID**

Вы можете указать нижний предел числу валидных значений в наблюдении, целое положительное число. Если валидных значений в наблюдении массива окажется меньше этого порога, то наблюдение не берется в обработку, и в выходящих из макроса переменных оно будет пусто (sysmis). Если MISSING=INCLUDE, пользовательские пропуски считаются валидными значениями.

По умолчанию NVALID=1 для всех TRANS, кроме Z. Для TRANS=Z дефолтное NVALID=2 (и NVALID=1 трактуется как NVALID=2).

**FLAG**

Если CAP задано, то FLAG=YES создает или обновляет в массиве переменную *NVALID#$*, показывающую число валидных значений в каждом наблюдении, на входе. Кроме того, FLAG=YES создает или обновляет в массиве переменную *CONSTA#$*, помечающую наблюдения-константы, на входе. Константы, равные нулю, помечаются кодом 0, а прочие константы кодом 1.

FLAG=YES нельзя при незаданной CAP. FLAG не реагирует на п/к NVALID.

***Особые режимы***

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

# МАКРОС !KO\_HRANK: РАНЖИРОВАНИЕ ПО ГОРИЗОНТАЛИ

Version 2, Nov 2020 (Version 1, Jul 2004). Tested on SPSS Statistics 20, 22, 25.

!KO\_hrank vars= *v1 to v10* /\*Числовые переменные (смежные, ч-з to), значения которых

/\*горизонтально ранжировать

/cap= *rnk* /\*Приставка (можно окавыченно) в имена выходящих переменных; если не задать,

/\*сами VARS будут ранжированы

/trans= RANK /\*Преобразование: ранги (RANK), нормальные баллы (NORMAL),

/\*процентильные группы (NTILES k)

/order= D /\*Направление ранжирования: по возрастающей (A, тж п/у), по убывающей (D)

/ties= /\*Обращение с одинаковыми значениями: ранги усреднять (MEAN, тж п/у),

/\*дать нижний ранг (LOW), дать верхний ранг (HIGH),

/\*сжать ранги (CONDENSE), дать последовательные ранги (GRADE)

/fraction= /\*При TRANS=NORMAL, способ оценки доли: BLOM (тж п/у), TUKEY, RANKIT, VW (см.)

/missing= /\*Брать user-missing значения в ранжирование (INCLUDE) или не брать (EXCLUDE, тж п/у)

/nvalid= *8* /\*Опционально: опустошить наблюдение с числом валидных значений меньше порога

/\*(укажите положит. число)

/flag= YES /\*Если задано CAP: посчитать валидные значения: YES или NO (тж п/у).

Минимум надо задать VARS, TRANS.

Переводит значения числовых переменных в ранги по горизонтали, т.е. ранжирует данные внутри наблюдений. Вы можете затребовать ранги как новые переменные или же заменить ими сами входящие переменные. Вы можете по-разному обойтись с совпадающими значениями (ties). Макрос также может пересчитать данные на нормальное распределение (нормальные баллы) или сделать разбивку данных на k равночисленных групп (процентильные группы). Этот макрос делает то же, что команда RANK, только делает это «внутри наблюдений», а не «внутри переменных».

В рабочем массиве должно не быть переменных *NVARS###* и *NVARS##$*.

ПРИМЕР 1.

!KO\_hrank vars= v1 to v10 /cap= rnk /trans= RANK.

* Данные переменных V1 – V10 ранжируются по горизонтали, образуя переменные с рангами RNK1 – RNK10.

***Подкоманды***

**VARS**

Укажите ч-з «to» числовые переменные, значения которых ранжировать по горизонтали. Это должны быть смежные в массиве данных переменные. Данные воспринимаются как количественные (не номинальные).

**CAP**

Приставка в имена выходящих переменных с рангами. Ее можно взять в кавычки/апострофы (они рекомендуются, если приставку вы оканчиваете на точку: “var.”). Выходящие переменные будут поименованы приставкой с индексом от 1 до число входящих переменных. Во входящем массиве должно не быть переменных с этими именами.

Если не задать или умолчать подкоманду CAP, то сами входящие переменные VARS будут выходящими: все значения в них будут заменены, а любые user-missing дефиниции отменены. Ярлыки значений останутся прежние, но станут неуместны, т.к. значения переменных будут новые.

**TRANS**

Укажите требуемое преобразование данных (которое будет сделано индивидуально внутри каждого наблюдения массива):

RANK - ранжировать значения – перевести их в ранги.

NORMAL - сделав ранжирование, перевести затем ранги в нормальные баллы (normal scores), т.е. в значения, какие были бы у входящих данных, будь они из стандартного нормального распределения.

NTILES *k* - сделав ранжирование, сгруппировать затем ранги по их величине в *k* приблизительно равночисленных групп. Это категоризация (квантизация, binning) вариационного ряда количественных данных методом процентильных групп. Укажите целое положительное число не ниже 2. Например, NTILES 4 припишет код 1 значениям до 25-го процентиля, код 2 значениям между 25-м и 50-м процентилем, код 3 значениям между 50-м и 75-м процентилем, код 4 значениям за 75-м процентилем. Ожидается, что *k* невелико – меньше, чем число уникальных значений в вариационном ряду, т.е. в наблюдении массива.

Все три преобразования делаются идентично тому (по тем же формулам), как их делает команда RANK (см. SPSS Statistics Algorithms).

По причинам техническим макрос не делает преобразования NORMAL и NTILES в случае TIES=GRADE.

Другой макрос, который может выполнить ту же задачу, что делает опция NTILES – это [!KO\_HBINEF](#_МАКРОС__!KO_HBINEF:). Последний использует несколько другой подход – прямое вычисление значений процентилей методом “Aempirical” (нежели ранжирование, делаемое в !KO\_HRANK). Результаты категоризации !KO\_HBINEF и !KO\_HRANK совпадают достаточно часто. !KO\_HBINEF несколько быстрее.

**TIES**

Как обращаться с одинаковыми (tied) значениями при ранжировании. Поскольку ранжирование делается при любом TRANS, задание TIES затрагивает все те виды преобразования.

MEAN - (тж п/у или незаданию) присвоить им их средний ранг.

LOW - присвоить им нижний их ранг.

HIGH - присвоить им верхний их ранг.

CONDENSE - присвоить им нижний ранг и сжать ранги: следующий ранг будет отличаться на 1 от этого (в меню Rank Cases этот метод обозначен как “Sequential ranks for unique values”).

GRADE - ties получают разные ранги, последовательные целые.

Иллюстрация этих вариантов (ранжирование по возрастающей):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Значение* | *3* | *6* | *6* | *7* | *9* |
| GRADE | 1 | 2 | 3 | 4 | 5 |
| MEAN | 1 | 2.5 | 2.5 | 4 | 5 |
| HIGH | 1 | 3 | 3 | 4 | 5 |
| LOW | 1 | 2 | 2 | 4 | 5 |
| CONDENSE | 1 | 2 | 2 | 3 | 4 |

Максимальный ранг для всех вариантов, кроме CONDENSE, всегда равен числу валидных значений в наблюдении.

**ORDER**

Направление ранжирования: по возрастающей, т.е. больший ранг – большему значению (ORDER=A, тж по умолчанию/незаданию) или по убывающей: больший ранг – меньшему значению (ORDER=D).

**FRACTION**

Эта подкоманда обслуживает TRANS=NORMAL. Она задает способ оценки, в непрерывном распределении, накопленной доли, соответствующей определенному рангу.

BLOM - (тж п/у или незаданию) способ Блома.

TUKEY - способ Тьюки.

RANKIT - способ ранкит.

VW - способ Ван-дер-Вердена.

**MISSING**

Как относиться к пользовательским пропущенным (user-missing) значениям во входящих переменных: не брать их в ранжирование, приняв за system-missing (MISSING=EXCLUDE, тж по умолчанию/незаданию) или взять, как валидные значения (MISSING=INCLUDE).

**NVALID**

Вы можете указать нижний предел числу валидных значений в наблюдении, целое положительное число. Если валидных значений в наблюдении массива окажется меньше этого порога, то наблюдение не берется в обработку, и в выходящих из макроса переменных оно будет пусто (sysmis). Если MISSING=INCLUDE, пользовательские пропуски считаются валидными значениями. По умолчанию NVALID=1.

**FLAG**

Если CAP задано, то FLAG=YES создает или обновляет в массиве переменную *NVALID#$*, показывающую число валидных значений в каждом наблюдении, на входе. Если MISSING=INCLUDE, пользовательские пропуски считаются валидными значениями.

FLAG=YES нельзя при незаданной CAP. FLAG не реагирует на п/к NVALID.

***Особые режимы***

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

# МАКРОС !KO\_HSORT: СОРТИРОВКА ИЛИ РАНДОМИЗАЦИЯ ПО ГОРИЗОНТАЛИ

Version 2, Dec 2020 (Version 1, Mar 2005). Tested on SPSS Statistics 20, 22, 25.

!KO\_hsort vars= *v1 to v10* /\*Числовые переменные (смежные, ч-з to), значения которых горизонтально

/\*сортировать или рандомизовать

/cap= *s\_*  /\*Приставка (можно окавыченно) в имена выходящих переменных; если не задать,

/\*сами VARS будут сортированы (рандомизованы)

/cases= FIRST /\*Сортировать (рандомизовать) независимо внутри всех наблюдений (ALL, тж п/у); или

/\*сортировать (рандомизовать) внутри только первого наблюдения, а остальные

/\*прихватить (FIRST)

/selvar= *sel=1* /\*Опционально: переменная для выбора наблюдений, затрагиваемых макросом:

/\*имя и условие

/order= /\*Сортировка или рандомизация: по возрастающей (A, тж п/у), по убывающей (D),

/\*рандомизация (R)

/missing= /\*Брать user-missing значения в обработку (INCLUDE) или не брать (EXCLUDE, тж п/у)

/misspos= ORIG /\*Если сортировка: пропущенные значения сортировать в конец (TAIL, тж п/у) или

/\*оставить их на месте (ORIG)

/redupl= /\*Если сортировка: повторения значений не удалять (LEAVE, тж п/у)

/\*или удалить (CROP).

Минимум надо задать VARS.

Сортирует или рандомизует значения числовых переменных по горизонтали, внутри наблюдений. Можно создать новые переменные или заменить сами входящие переменные. Часть наблюдений можно исключить из этой процедуры перестановки значений, а из взятых наблюдений можно по желанию сортировать (рандомизовать) только одно, остальные же прихватить с ним. Макрос не делает вложенной сортировки (т.е. сортировку наблюдения внутри одинаковых значений другого наблюдения).

Алгоритм сортировки, употребляемый макросом !KO\_HSORT, умеренно скор. Для сортировки по горизонтали большого множества длинных векторов (т.е. многих наблюдений и большого числа переменных VARS) используйте более скорый макрос [!KO\_HQSORT](#_МАКРОС_!HQSORT:_БЫСТРАЯ).

В рабочем массиве должно не быть переменных *NVARS###* и *NVARS##$*.

ПРИМЕР 1.

!KO\_hsort vars= v18.1 to v18.9 /order= A /cap= 's18.'.

!KO\_hsort vars= v18.1 to v18.9 /order= A /cap= 's18\_' /cases= FIRST.

* В первом пуске значения переменных V18.1 – V18.9 сортируются слева направо по возрастающей. Так делается для каждого наблюдения массива данных. Выходящие переменные: S18.1 – S18.9.
* Во втором пуске в тех же переменных сортируются значения только первого наблюдения массива данных. Остальные наблюдения переставляют свои данные вослед первому наблюдению. Выходящие переменные: S18\_1 – S18\_9.

***Подкоманды***

**VARS**

Укажите ч-з «to» числовые переменные, значения которых сортировать (или рандомизовать) по горизонтали. Это должны быть смежные в массиве данных переменные.

**CAP**

Приставка в имена выходящих переменных с переставленными значениями. Ее можно взять в кавычки/апострофы (они рекомендуются, если приставку вы оканчиваете на точку: “var.”). Выходящие переменные будут поименованы приставкой с индексом от 1 до число входящих переменных. Во входящем массиве должно не быть переменных с этими именами.

Если не задать или умолчать подкоманду CAP, то сами входящие переменные VARS будут выходящими: значения в них будут переставлены, а любые user-missing дефиниции отменены (но: см. п/к SELVAR). Ярлыки значений останутся прежние, но могут стать неуместны, т.к. значения переменных будут горизонтально переставлены.

**SELVAR**

По умолчанию макрос затрагивает все наблюдения массива данных. Данной подкомандой можно указать одну существующую в массиве переменную не из списка VARS, выбирающую наблюдения для обработки макросом. После имени переменной задайте одно простое условие, состоящее из отношенческого оператора (=, <>, >, <, >=, <=, EQ, NE, GT, LT, GE, LE) и валидного значения (либо имени иной переменной на месте значения). Примеры допустимых заданий: SELVAR= *REGION=2*; или: SELVAR= *HEIGHT>26.8*; или: SELVAR= *X<>Y*. Вы вправе употребить скобки, если хотите например: SELVAR= *(REGION=2)*.

Макросом обработаются только те наблюдения, которые удовлетворят условию. Кроме того: в переменной, указываемой в п/к SELVAR, если есть пропущенные значения, то эти наблюдения станут пустые (system missing) в выходящих переменных.

Если п/к SELVAR задана, а CAP нет, то макрос не станет отменять user-missing дефиниции в переменных VARS на выходе и предупредит об этом.

ПРИМЕР 2.

!KO\_hsort vars= q4 to q21 /selvar= region<>5.

!KO\_hsort vars= q4 to q21 /selvar= region=5 /missing= INCLUDE /redupl= CROP.

* Первая команда сортирует значения наблюдений, которые удовлетворяют “REGION не есть 5”.
* Вторая команда сортирует значения наблюдений, которые удовлетворяют “REGION есть 5”. При этом пользовательские пропуски принимаются за валидные данные, повторы значений отсеиваются.

**CASES**

Эта подкоманда указывает, какие из затрагиваемых макросом наблюдений *переставляемые* – т.е. их значения будут слева направо сортированы (или рандомизованы), – а какие *прихвачены* в перестановку. Одно из двух ключевых слов:

ALL - (тж по умолчанию/незаданию п/к) переставляемыми, независимо друг от друга, будут все отобранные наблюдения массива: во всех них значения будут слева направо сортированы (рандомизованы). Прихваченных наблюдений нет.

FIRST - переставляемое наблюдение одно – первое в массиве, а остальные затрагиваемые наблюдения будут прихвачены в перестановку. Другими словами, это перестановка *переменных* (столбцов данных только, без перестановки имен переменных) вослед переупорядочиванию значений первого наблюдения массива.

При сортировке (ORDER= A или D) в условиях CASES=FIRST, если у первого наблюдения есть пропущенные значения, в выходящих данных под ними будут также лишь пропущенные, system-missing значения. Иначе говоря, пропущенное сортируемое значение не может прихватить с собой значения в прочих наблюдениях.

**ORDER**

Вид перестановки: сортировать слева направо по возрастанию значений (ORDER=A, тж по умолчанию/незаданию), сортировать слева направо по убыванию значений (ORDER=D), или рандомизовать порядок значений (ORDER=R)[[1]](#footnote-1).

Для сортировки (ORDER= A или D) макрос использует устойчивый Selection-алгоритм сортировки. «Устойчивый» алгоритм сортировки означает, что порядок среди одинаковых (повторяющихся) значений сортировка сохранит. Эта устойчивость проявляется в условиях CASES=FIRST. К примеру, если в первом наблюдении переменная *X*=2 и переменная *Y*=2, и *X* находится в массиве левее, чем *Y*, то устойчивый сортировочный алгоритм поставит содержимое столбца *X* гарантированно левее содержимого столбца *Y*.

**MISSING**

Как относиться к пользовательским пропущенным (user-missing) значениям во входящих переменных в *переставляемых* наблюдениях: считать их пропусками, как system-missing (MISSING=EXCLUDE, тж по умолчанию/незаданию) или взять, как валидные значения (MISSING=INCLUDE).

**MISSPOS**

Эта подкоманда не действует при ORDER=R. Она позволяет затребовать, чтобы пропущенные значения в *переставляемых* наблюдениях остались на своих местах (MISSPOS=ORIG), а не сортировались в хвост, после валидных значений (MISSPOS=TAIL, тж. по умолчанию/незаданию). Иначе говоря, при MISSPOS=ORIG ячейка с пропущенным значением не сможет оказаться содержащим на выходе валидное значение, такая ячейка будет system-missing.

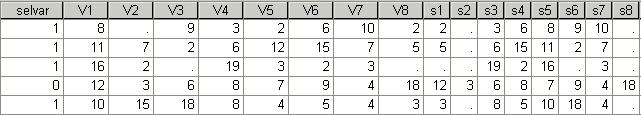
**REDUPL**

Эта дополнительная опция (не действующая при ORDER=R) – отсеивание повторяющихся значений в ходе сортировки. При REDUPL=CROP, если у *переставляемого* наблюдения некое значение встречается более раза, в выходящих переменных оно будет встречаться у него лишь раз. По умолчанию/незаданию и при REDUPL=LEAVE выброса значений-повторов нет.

При сортировке CASES=FIRST и REDUPL=CROP столбец прихваченных значений, расположенный в выходящих данных под значением, которое было купировано (лишено повторов), отвечает *последней* (правейшей) встрече этого значения во входящих данных.

ПРИМЕР 3.

!KO\_hsort vars= v1 to v8 /order= A /cap= s /cases= FIRST /selvar= selvar=1 /misspos= ORIG /redupl= CROP.



* Значения переменных V1 – V8 заказано сортировать, слева направо по возрастающей. Сортироваться будет только первое наблюдение, а остальные – прихвачены в эту сортировку (CASES=FIRST). Выходящие данные – новые переменные S1 – S8.
* Согласно переменной SELVAR, 4-е наблюдение вообще исключено из процесса: в переменных S1 – S8 его данные поэтому не перемещены по сравнению с входящими переменными.
* У сортируемого (первого) наблюдения есть пропущенное значение во второй переменной. Поскольку MISSPOS=ORIG, в сортированных данных этого наблюдения пропуск остался на своем месте, нежели отошел в конец.
* В конце (S8) пропуск у первого наблюдения возник по другой причине: заказано было купировать дублирующиеся значения (REDUPL=CROP). Значение 2 дважды встречалось в сортируемом наблюдении: теперь оно осталось одно.
* В переменной S1 купированное значение, 2, прихватило с собой данные, бывшие под последнем (вторым) значением 2 во входящих данных (V8, а не V5).
* Пропущенные значения у первого наблюдения не сопровождаются какими-либо прихваченными значениями: столбцы S2 и S8 пусты.

***Особые режимы***

Макрос не реагирует на взвешенность, расщепленность массива данных. Он игнорирует временные (под TEMPORARY) операции. Макрос не слушается команды FILTER или USE: для отбора наблюдений в процедуру используйте подкоманду SELVAR макроса. С оглядкой пускайте перед макросом команды, удаляющие из массива наблюдения (такие как SELECT IF и SAMPLE), когда полагаете CASES=FIRST, т.к. может статься, что первым в массиве будет не то наблюдение, о каком вы думали.

# МАКРОС !KO\_HQSORT: БЫСТРАЯ СОРТИРОВКА ПО ГОРИЗОНТАЛИ

Version 2, Dec 2020 (Version 1, Dec 2007). Tested on SPSS Statistics 20, 22, 25.

!KO\_hqsort vars= *v1 to v1000* /\*Числовые переменные (смежные, список ч-з to), значения которых

/\*горизонтально сортировать

/cap= *s\_* /\*Приставка (можно окавыченно) в имена выходящих переменных; если не задать,

/\*сами VARS будут сортированы

/selvar= /\*Опционально: переменная для выбора наблюдений, затрагиваемых макросом:

/\*имя и условие

/order= /\*Сортировка по возрастающей (A, тж п/у) или по убывающей (D)

/redupl= /\*Повторения значений не удалять (LEAVE, тж п/у)

/\*или удалить (CROP).

Минимум надо задать VARS.

Сортирует значения числовых переменных по горизонтали, внутри наблюдений. Можно создать новые переменные или заменить сами входящие переменные. Часть наблюдений можно исключить из сортировки. Макрос не делает вложенной сортировки (т.е. сортировку наблюдения внутри одинаковых значений другого наблюдения).

Макрос использует Quicksort-алгоритм сортировки (с pivot-значением в середине подвектора), который скор и годится для большого числа длинных векторов (т.е. многих наблюдений и большого числа переменных VARS). Он даст тот же результат, что макрос [!KO\_HSORT](#_МАКРОС_!HSORT:_СОРТИРОВКА). Однако !KO\_HQSORT не имеет опций, которые есть в !KO\_HSORT, а именно вариантов ORDER=R, CASES=FIRST, MISSPOS=ORIG. !KO\_HQSORT сортирует независимо друг от друга все (взятые в сортировку) наблюдения массива. Пропуски отсортировываются в хвост. !KO\_HQSORT не имеет подкоманды MISSING: если вы хотите включить в сортировку пользовательские пропуски, снимите с входящих переменных user-missing дефиниции сами.

В рабочем массиве должно не быть переменных *NVARS###* и *NVARS##$*.

ПРИМЕР 1.

!KO\_hqsort vars= f1 to f1042 /order= A /selvar= sex=1.

* Значения переменных F1 – F1042 сортируются слева направо по возрастающей. Так делается для каждого наблюдения массива данных, для которого SEX=1. Выходящие переменные: сами F1 – F1042.

***Подкоманды***

**VARS**, **CAP**, **SELVAR**

Эти подкоманды тождественны таковым макроса [!KO\_HSORT](#_МАКРОС_!HSORT:_СОРТИРОВКА) (см.).

**ORDER**

Направление сортировки: сортировать слева направо по возрастанию значений (ORDER=A, тж по умолчанию/незаданию) или по убыванию значений (ORDER=D).

**REDUPL**

Эта дополнительная опция – отсеивание повторяющихся значений в ходе сортировки. При REDUPL=CROP, если у наблюдения некое значение встречается более раза, в выходящих переменных оно будет встречаться у него лишь раз. По умолчанию/незаданию и при REDUPL=LEAVE выброса значений-повторов нет.

***Особые режимы***

Макрос не реагирует на взвешенность, расщепленность массива данных. Он игнорирует временные (под TEMPORARY) операции. Макрос не слушается команды FILTER или USE: для отбора наблюдений в процедуру используйте подкоманду SELVAR макроса.

# МАКРОС !KO\_HBINEF: КАТЕГОРИЗАЦИЯ ПО ГОРИЗОНТАЛИ (РАВНОЧАСТОТНЫЕ ГРУППЫ)

Version 1, Nov 2021. Tested on SPSS Statistics 20, 22, 26.

!KO\_hbinef vars= *v1 to* *v20* /\*Числовые переменные (смежные, ч-з to), значения которых

/\*горизонтально категоризовать

/cap= *bin* /\*Приставка (можно окавыченно) в имена выходящих переменных; если не задать,

/\*сами VARS будут категоризованы

/k= *4* /\*Число групп (категорий)

/order= A /\*Направление сортировки: по возрастающей (A), по убывающей (D);

/\*если сортировка в наблюдениях уже сделана, то далее можно кл слово SORTED

/incp= SEN /\*Точку насечки включить: в младшую (JUN) или в старшую (SEN) категорию

/trunc= NO /\*Усечение уровня процентиля: YES [d] (тж п/у) или NO

/missing= /\*Брать user-missing значения (INCLUDE) или не брать (EXCLUDE, тж п/у)

/nvalid= /\*Опционально: опустошить наблюдение с числом валидных значений меньше порога

/\*(укажите положит. число)

/cutp= *cp#* /\*Опционально: создать переменные с точками насечки (укажите приставку)

/freq= *fr#* /\*Опционально: создать переменные с частотами в категориях (укажите приставку)

/flag= YES /\*Если задано CAP: посчитать валидные значения: YES или NO (тж п/у).

Минимум надо задать VARS, K, ORDER, INCP.

Категоризует (квантизует) значения числовых (количественных) переменных по горизонтали, внутри наблюдений. Можно создать новые переменные или заменить сами входящие переменные. Создаваемые категоризацией порядковые категории – это равночастотные процентильные группы.

Другой макрос, который тоже может создавать равночастотные процентильные группы – [!KO\_HRANK](#_МАКРОС_!HRANK:_РАНЖИРОВАНИЕ). Сравнение см. ниже.

В рабочем массиве должно не быть переменных *NVARS###* и *NVARS##$*.

ПРИМЕР 1.

!KO\_hbinef vars= v1 to v50 /cap= bin /k= 4 /order= A /incp= JUN.

* Данные переменных V1 – V50 категоризуются по горизонтали, образуя переменные с категориями, BIN1 – BIN50.
* Категоризация – на 4 равнонаполненнные процентильные группы, т.е. квартильные группы (K=4). Категоризация идет от низких значений к высоким (order=A). Таким образом, макрос припишет код 1 значениям до 25-го процентиля (в распределении значений внутри наблюдения), код 2 значениям между 25-м и 50-м процентилем, код 3 значениям между 50-м и 75-м процентилем, код 4 значениям за 75-м процентилем. Так делается отдельно и независимо в каждом наблюдении массива.
* Если значение точно равно процентилю (точке насечки), ему будет приписан меньший код из двух конкурентных (INCP=JUN). Например, значению, совпадающему с 25-м процентилем, будет придан код 1 (а не код 2).

**Алгоритм**. Следующее делается независимо в каждом наблюдении.

1. Сортировка. *n* валидных данных в наблюдении внутренне сортируются возрастающе, после чего им присваиваются порядковые индексы *i* = 1, 2, ..., *n*.
2. Точки насечения. *k*-1 точек насечения на приблизительно равночастотные группы – это значения процентилей, отвечающие процентильным уровням (процентильным рангам) 1/*k*\*100%, 2/*k*\*100%, ..., (*k*-1)/*k*\*100%. Например, при *k*=4 точки насечения станут значениями процентилей: 25%-го, 50%-го, 75%-го. Значения процентилей определяются методом “Empirical method with averaging” (метод AEMPIRICAL в SPSS-команде EXAMINE – см. “Command Syntax Reference” и “SPSS Statistics Algorithms”).
3. Конденсация. Повторы точек насечения, если есть, удаляются. Поэтому в некоторых случаях точек насечения остается меньше, чем *k*-1. Конденсация не скажется на группировании данных; она делается ради того, чтобы коды групп были непрерывными 1, 2, ..., без лакун.
4. Приписание кодов данным. Пусть число точек насечения после (3) есть *p*, и их значения есть *Q1*, *Q2*, ..., *Qp*. Тогда, для *c* = 1, 2, ..., *p*: если значение данного < (при INCP=SEN) или ≤ (при INCP=JUN) чем *Qc*, то данному присваиваем код *c*, и переходим к другому данному.

Таким образом, данные получают коды (категории) от 1 до *p* (с идеалом *p*=*k*), меньшие значения получают меньший код. *p* категорий, насколько возможно, равнонаполненны данными. Если ORDER=D (а не A), то в (1) данные сортируются убывающе, а в (4) вместо “<” и “≤” используются “>” и “≥”; в итоге большие значения получают меньший код.

В редких случаях, когда число уникальных значений в наблюдении мало для заданного *k*, коды могут выйти не последовательными, несмотря на (3); например, возможна кодировка 1 и 3 (вместо 1 и 2).

* Категоризация, осуществляемая !KO\_HBINEF, совпадает по результату с категоризацией, осуществляемой меню-основанной процедурой Visual Binning, опция – “Equal Percentiles” (проверено на верс. 26). Но Visual Binning категоризует данные внутри переменных, а !KO\_HBINEF делает это внутри наблюдений.
* SPSS команда OPTIMAL BINNING со спецификацией METHOD=EQUALFREQ дает результаты, как правило близкие к Visual Binning / !KO\_HBINEF (проверено на верс. 26). OPTIMAL BINNING использует похожий алгоритм, но жестче следит за последовательностью (конденсированностью) итоговых кодов.
* Макрос [!KO\_HRANK](#_МАКРОС_!HRANK:_РАНЖИРОВАНИЕ) со спецификацией /TRANS= NTILES *k* тоже выполняет, как и !KO\_HBINEF, категоризацию на равнонаполненные процентильные группы, но использует несколько иной алгоритм, основанный на рангах (а не на вычислении значений процентилей), и где результат категоризации зависит от способа обработки связок (ties). Результаты категоризации !KO\_HBINEF и !KO\_HRANK совпадают достаточно часто. !KO\_HBINEF несколько быстрее.

***Подкоманды***

**VARS**

Укажите ч-з «to» числовые переменные, значения которых категоризовать по горизонтали. Это должны быть смежные в массиве данных переменные. Данные воспринимаются как количественные (не номинальные).

**CAP**

Приставка в имена выходящих переменных с кодами категорий. Ее можно взять в кавычки/апострофы (они рекомендуются, если приставку вы оканчиваете на точку: “var.”). Выходящие переменные будут поименованы приставкой с индексом от 1 до число входящих переменных. Во входящем массиве должно не быть переменных с этими именами.

Если не задать или умолчать подкоманду CAP, то сами входящие переменные VARS будут выходящими: все значения в них будут заменены, а любые user-missing дефиниции отменены. Ярлыки значений останутся прежние, но станут неуместны, т.к. значения переменных будут новые.

**K**

Создать это число равночастотных групп (категорий). Укажите целое не меньше 2. Ожидается, что *k* невелико – меньше, чем число уникальных значений в вариационном ряду, т.е. в наблюдении массива.

**ORDER**

Направление сортировки данных, которую макрос сделает внутренне в каждом наблюдении. Сортировка значений делается в первую очередь, предшествуя вычислению точек насечения. Она определит порядок итоговых кодов. Если сортировка возрастающая (A), то группа с меньшим кодом будет содержать низшие значения данных. Если сортировка убывающая (D), то группа с меньшим кодом будет содержать высшие значения данных. ORDER=D имеет тот же эффект, как если входящим данным поменять знак и потом использовать ORDER=A.

В том случае, если данные к каждом наблюдении у вас уже сортированы (в порядке A или D), вы можете добавить кл. слово SORTED после кл. слова A или D. Это сэкономит время: макрос не станет сортировать входящие данные. Важно: в сортированных данных все пропуски (system-, user-) должны быть в хвосте (в правой части) наблюдения.

Сделаем замечание, что ORDER=D – не то же самое и не всегда дает тот же эффект, что поменять порядок кодов на выходе, т.е. перевернуть шкалу присвоенных кодов (Reverse scale: 1 2 3... в ...3 2 1), что делает, например, Visual Binning. Если пользователь хочет перевернуть итоговую шкалу, он может это сделать самостоятельно.

**INCP**

К какую из двух смежных групп включить значение, которое равно точке насечения между ними. INCP=JUN помещает это значение в “младшую” группу, т.е. группу с меньшим *кодом*. INCP=SEN помещает это значение в “старшую” группу, т.е. группу с бо́льшим *кодом*. Короче, JUN это *cut-point]*, а SEN это *]cut-point*.

**TRUNC**

Эта подкоманда определяет точность вычисления процентильных уровней (процентильных рангов). Она потенциально сказывается на значениях процентилей, следовательно, на тонконюансных смещениях среди точек насечения. Укажите TRUNC=YES *d* (где *d* – цифра от 1 до 6, означает оставляемое число десятичных цифр в дробном значении) либо TRUNC=NO.

Умолчание/незадание п/к, как и TRUNC=YES (без параметра *d*), соответствуют заданию TRUNC= YES 4. Всякий процентильный уровень будет усечен до 4 десятичных цифр. Например, при K=3 процентильные уровни будут: 33.3333% и 66.6666%. Если *d* поставить равным 2, эти уровни будут 33.33% и 66.66%. Если TRUNC=NO, то усечения не делается и дробные процентильные уровни предстают с машинной точностью (15-16 десятичных цифр).

**MISSING**

Как относиться к пользовательским пропущенным (user-missing) значениям во входящих переменных: не брать их в категоризацию, приняв за system-missing (MISSING=EXCLUDE, тж по умолчанию/незаданию) или взять, как валидные значения (MISSING=INCLUDE).

**NVALID**

Вы можете указать нижний предел числу валидных значений в наблюдении, целое положительное число. Если валидных значений в наблюдении массива окажется меньше этого порога, то наблюдение не берется в обработку, и в выходящих из макроса переменных оно будет пусто (sysmis). Если MISSING=INCLUDE, пользовательские пропуски считаются валидными значениями. По умолчанию NVALID=1.

**CUTP**

Вы можете затребовать создать новые переменные, показывающие точки насечения (после конденсации – см. «Алгоритм» выше). Предложите приставку для имен этих переменных. Ее можно взять в кавычки/апострофы (они рекомендуются, если приставку вы оканчиваете на точку: “var.”).

**FREQ**

Вы можете затребовать создать новые переменные, показывающие частоты в получившихся группах (категориях). Предложите приставку для имен этих переменных. Ее можно взять в кавычки/апострофы (они рекомендуются, если приставку вы оканчиваете на точку: “var.”).

**FLAG**

Если CAP задано, то FLAG=YES создает или обновляет в массиве переменную *NVALID#$*, показывающую число валидных значений в каждом наблюдении, на входе. Если MISSING=INCLUDE, пользовательские пропуски считаются валидными значениями.

FLAG=YES нельзя при незаданной CAP. FLAG не реагирует на п/к NVALID.

***Особые режимы***

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

# МАКРОС !KO\_HBINEI: КАТЕГОРИЗАЦИЯ ПО ГОРИЗОНТАЛИ (РАВНОИНТЕРВАЛЬНЫЕ ГРУППЫ)

Version 1, Nov 2021. Tested on SPSS Statistics 20, 22, 26.

!KO\_hbinei vars= *v1 to v20* /\*Числовые переменные (поименно и/или ч-з to), значения которых

/\*горизонтально категоризовать

/cap= *bin* /\*Приставка (можно окавыченно) в имена выходящих переменных; если не задать,

/\*сами VARS будут категоризованы

/k= *4* /\*Число групп (категорий)

/order= A /\*Направление: по возрастанию (A, тж п/у) или убыванию (D)

/sw= /\*Начальное значение и ширина интервала; или AUTO (тж п/у): определить

/\*автоматически

/incp= JUN /\*Точку насечки включить: в младшую (JUN) или в старшую (SEN) категорию

/outcodes= *-999 999* /\*Если задано SW: два кода для значений вне рабочего диапазона

/midinter= /\*Коды в виде срединтервальных значений (YES) или порядковых целых (NO, тж п/у)

/missing= /\*Брать user-missing значения (INCLUDE) или не брать (EXCLUDE, тж п/у)

/nvalid= /\*Опционально: опустошить наблюдение с числом валидных значений меньше порога

/\*(укажите положит. число)

/cutp= *c#* /\*Опционально: создать переменные с точками насечки (укажите приставку)

/freq= *f#* /\*Опционально: создать переменные с частотами в категориях (укажите приставку)

/flag= YES /\*Если задано CAP: посчитать валидные значения: YES или NO (тж п/у).

Минимум надо задать VARS, K, INCP.

Категоризует (квантизует) значения числовых (количественных) переменных по горизонтали, внутри наблюдений. Можно создать новые переменные или заменить сами входящие переменные. Создаваемые категоризацией порядковые категории – это равноинтервальные группы.

В рабочем массиве должно не быть переменных *NVARS###* и *NVARS##$*.

ПРИМЕР 1.

!KO\_hbinei vars= v1 to v50 /cap= bin /k= 4 /order= A /incp= JUN.

* Данные переменных V1 – V50 категоризуются по горизонтали, образуя переменные с категориями, BIN1 – BIN50.
* Категоризация – на 4 равноинтервальные группы. Категоризация идет от низких значений к высоким (order=A). По умолчанию п/к SW начало 1-го интервала и ширина интервала определяются автоматически: начало это минимальное наблюдаемое значение, и наблюдаемый в данных диапазон делится на K=4 интервала. Так делается отдельно и независимо в каждом наблюдении массива.
* Если значение точно равно границе между интервалами (точке насечки), ему будет приписан меньший код из двух конкурентных (INCP=JUN). Например, значению, совпадающему с границей 1-го и 2-го интервалов, будет придан код 1 (а не код 2).

***Подкоманды***

**VARS**

Укажите числовые переменные (минимум две), значения которых категоризовать по горизонтали. Список поименно и/или ч-з «to». Если переменных очень много, рекомендуется использовать «to»: это может сэкономить время исполнения. Данные воспринимаются как количественные (не номинальные).

**CAP**

Приставка в имена выходящих переменных с кодами категорий. Ее можно взять в кавычки/апострофы (они рекомендуются, если приставку вы оканчиваете на точку: “var.”). Выходящие переменные будут поименованы приставкой с индексом от 1 до число входящих переменных. Во входящем массиве должно не быть переменных с этими именами.

Если не задать или умолчать подкоманду CAP, то сами входящие переменные VARS будут выходящими: все значения в них будут заменены, а любые user-missing дефиниции отменены. Ярлыки значений останутся прежние, но станут неуместны, т.к. значения переменных будут новые.

**K**

Создать это число равноинтервальных групп (категорий). Укажите целое не меньше 2. Ожидается, что *k* невелико – меньше, чем число уникальных значений в вариационном ряду, т.е. в наблюдении массива.

**ORDER**

Направление интервализации данных (т.е. насечения шкалы значений на равные интервалы), которую макрос сделает в каждом наблюдении. Она определит порядок итоговых кодов. Если интервализация возрастающая (A), то группа с меньшим кодом будет содержать низшие значения данных. Если сортировка убывающая (D), то группа с меньшим кодом будет содержать высшие значения данных.

**SW**

Подкоманда, задающая параметры интервализации шкалы значений – начало насечения на *k* равных интервалов и ширину интервала.

AUTO - (тж по умолчанию/незаданию) определяет параметры для каждого наблюдения данных автоматически. Начало это минимальное (*min*, при ORDER=A) или максимальное (*max*, при ORDER=D) валидное значение в наблюдении. Насекаемый диапазон совпадает с наблюдаемым диапазоном валидных данных в наблюдении, *max-min*. Ширина интервала = *(max-min)/k*. При ORDER=A насечение идет от *min* к *max*, а при ORDER=D – наоборот. При SW=AUTO все валидные значения в наблюдении окажутся в какой-либо из *k* категорий.

*Start* *Width* - задание начала и ширины интервала вручную. Укажите два числа; второе (ширина) должно быть положительным. Первое (начало), ожидается, будет из области низших значений при ORDER=A, поскольку насечение пойдет от него вверх. И ожидается, что начало будет из области высших значений при ORDER=D, поскольку насечение пойдет от него вниз. При ручном задании SW не обязательно будет, что все валидные значения в наблюдении попадут в одну из *k* категорий; не попавшие ни в одну из категорий значения будут перекодированы в коды OUTCODES (см.).

Любой или оба параметра *Start* и *Width* вы можете задать именем переменной, а не только числом. Это должны быть переменные массива, содержащие нужные числовые значения. Так вы можете задать разные параметры разным наблюдениям. Ширина должна быть всегда положительным значением.

Если переменная со значениями *Start* содержит пропуски, то эти наблюдения в итоге работы сохранят свои входящие валидные значения: они не будут категоризованы никак (однако user-missing значения в них будут обращены в *sysmis*, если MISSING=EXCLUDE). Этим обстоятельством можно пользоваться, если нужно, чтобы макрос не затрагивал некоторые наблюдения массива.

ПРИМЕР 2.

!KO\_hbinei vars= v1 to v50 /cap= cat /k= 5 /order= D /sw= ptile90 2.5 /incp= JUN /outcodes= 999 99.

* Данные переменных V1 – V50 категоризуются по горизонтали, образуя переменные с категориями, CAT1 – CAT50.
* Категоризация – на 5 равноинтервальных групп. Она идет (группы 1, 2, 3, 4, 5) от высоких значений к низким (order=D), начиная со значения, содержащегося в переменной PTILE90 и с интервалом шириной 2.5.
* Значения, превышающие PTILE90, будут перекодированы в 999, а те, что слишком малы, чтобы попасть в последний (5-й) интервал, будут перекодированы в 99 (п/к OUTCODES).
* Если значение точно равно границе между смежными интервалами, оно отходит к группе с меньшим кодом (INCP=JUN).

**INCP**

К какую из двух смежных групп включить значение, которое равно точке насечения между ними. INCP=JUN помещает это значение в “младшую” группу, т.е. группу с меньшим *кодом*. INCP=SEN помещает это значение в “старшую” группу, т.е. группу с бо́льшим *кодом*. Короче, JUN это *cut-point]*, а SEN это *]cut-point*.

При ручном задании параметров SW эта инструкция касается и границ диапазона насечения, т.е. точек *Start* и *cut-point\_k*. Поэтому при INCP=JUN значение, равное *Start*, не попадет в группу 1, оно окажется вне категорий и будет кодировано первым кодом, указанным в OUTCODES. При INCP=SEN значение, равное *cut-point\_k*, не попадет в группу k, оно окажется вне категорий и будет кодировано вторым кодом, указанным в OUTCODES. А вот при SW=AUTO инструкция INCP касается только точек насечения, лежащих внутри от границ *Start* и *cut-point\_k*. Так что значение, равное *Start*, всегда войдет в группу 1, и значение равное *cut-point\_k*, всегда войдет в группу k.

**OUTCODES**

Подкоманда игнорируется при SW=AUTO. Укажите два значения (кода), в которые надо перекодировать всякое валидное значение, не попадающее в насеченный диапазон. Первый код получат значения, лежащие «до» *Start*. Второй код получат значения, лежащие «после» *cut-point\_k*. При ORDER=A «до» это низкие значения, а «после» это высокие значения; при ORDER=D – наоборот. Коды могут быть и одинаковы.

Любое или оба значения вы можете задать именем переменной, не только числом. Это должны быть переменные массива, содержащие нужные числовые коды. Так вы можете задать разные коды разным наблюдениям. Вы можете также использовать на месте значения кл. слово $SYSMIS (системный пропуск).

**MIDINTER**

По умолчанию и при MIDINTER=NO приписываемые группам коды категорий это целые 1, 2, ..., *k*. При MIDINTER=YES коды категорий это значения, равные серединам интервалов. Например, группа 2 будет кодирована не как 2, как *(cut-point\_2 + cut-point\_1) / 2*.

**MISSING**

Как относиться к пользовательским пропущенным (user-missing) значениям во входящих переменных: не брать их в категоризацию, приняв за system-missing (MISSING=EXCLUDE, тж по умолчанию/незаданию) или взять, как валидные значения (MISSING=INCLUDE).

**NVALID**

Вы можете указать нижний предел числу валидных значений в наблюдении, целое положительное число. Если валидных значений в наблюдении массива окажется меньше этого порога, то наблюдение не берется в обработку, и в выходящих из макроса переменных оно будет пусто (sysmis). Если MISSING=INCLUDE, пользовательские пропуски считаются валидными значениями. По умолчанию NVALID=1.

**CUTP**

Вы можете затребовать создать новые переменные, показывающие точки насечения на равные интервалы. Предложите приставку для имен этих переменных. Ее можно взять в кавычки/апострофы (они рекомендуются, если приставку вы оканчиваете на точку: “var.”). Начальная точка интервализации показывается как точка насечения с индексом 0.

**FREQ**

Вы можете затребовать создать новые переменные, показывающие частоты в получившихся группах (категориях). Предложите приставку для имен этих переменных. Ее можно взять в кавычки/апострофы (они рекомендуются, если приставку вы оканчиваете на точку: “var.”).

**FLAG**

Если CAP задано, то FLAG=YES создает или обновляет в массиве переменную *NVALID#$*, показывающую число валидных значений в каждом наблюдении, на входе. Если MISSING=INCLUDE, пользовательские пропуски считаются валидными значениями.

FLAG=YES нельзя при незаданной CAP. FLAG не реагирует на п/к NVALID.

***Особые режимы***

Макрос не реагирует на взвешенность, фильтрованность (FILTER, USE), расщепленность файла данных. Он игнорирует временные (под TEMPORARY) преобразования. Если вы хотите, чтобы макрос не затрагивал некоторые наблюдения массива, задайте параметр *Start* как переменную с пропусками (чит. в подкоманде SW).

# МАКРОС !KO\_HFREQ: ЧАСТОТЫ, ПО ГОРИЗОНТАЛИ

Version 1, Dec 2018. Tested on SPSS Statistics 20, 22, 26.

!KO\_hfreq vars= *v1 v5 to v50* /\*Числовые переменные (поименно и/или ч-з to), частоту значений

/\*которых подсчитывать горизонтально

/values= *-1 THRU 10* /\*Опционально: среди каких именно значений подсчитывать

/discret= RND *1* /\*Как дискретизировать входящие значения: RND d или TRUNC d (п/у RND 0)

/cap= f# /\*Приставка (можно окавыченно) в имена выходящих переменных с частотами

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

/\*или индекс (INDEX).

Минимум надо задать VARS, CAP.

Подсчитывает частоты значений по горизонтали, внутри наблюдений. Результат – новый безымянный массив, наблюдения в котором это наблюдения входящего массива, а переменные соответствуют значениям входящих данных; значения этих переменных – частоты.

Переменная *NTOTAL#$* в выходящем массиве содержит сумму частот. Эта переменная не будет создана, если во входящих данных нет пропусков и все существующие во входящих данных валидные значения были обработаны макросом (п/к VALUES позволяет взять в обработку не все существующие значения).

Макрос полагает user-missing значения пропусками и не подсчитывает частоты для них. Макрос не изменяет никак данные во входящем массиве.

ПРИМЕР 1.

!KO\_hfreq vars= v1 to v50 /cap= f#.

* Отдельно в каждом наблюдении массива подсчитываются частоты разных значений, существующих в столбцах V1 – V50. Если значения нецелые, они внутренне округляются до целых (см. п/к DISCRET).
* Будет создан новый массив. Наблюдения в нем это наблюдения входящего, а переменные (столбцы) в нем это разные существующие во входящих данных значения (само значение отражено в имени переменной). Данные в нем – это частоты.

***Подкоманды***

**VARS**

Укажите числовые переменные (минимум одна), частоты значения в которых подсчитывать по горизонтали. Список поименно и/или ч-з «to». Если переменных очень много, рекомендуется использовать «to»: это может сэкономить время исполнения.

**VALUES**

По умолчанию макрос подсчитывает частоту у всякого встреченного валидного значений. Здесь вы можете указать интересующий вас диапазон значений или их список. Тогда подсчет частот будет только у валидных значений, попадающих в этот диапазон/список. Для указания диапазона используйте следующие ключевые слова (принятые в SPSS-командах COUNT и RECODE): THRU, LO (или LOWEST), HI (или HIGHEST). Например, 0 6 THRU HI будет означать: диапазон от 6 и выше и еще значение 0.

ПРИМЕР 2.

!KO\_hfreq vars= v1 to v50 /cap= f# /values= -3 THRU 3.

* Данные переменных V1 – V50 это какие-то значения от -10 до 20. Но макрос будет интересоваться частотами только значений, находящихся в диапазоне от -3 до 3, так что для каждого уникального значения лишь из этого диапазона будут выданы частоты.

**DISCRET**

Эта подкоманда играет роль, если во входящих данных есть континуальные или дробные валидные значения, не отсеянные п/к VALUES. Вы должны указать макросу, округлять или усекать их прежде подсчета частот, и с какой точностью это делать. Укажите, соответственно, DISCRET= RND *d* либо DISCRET= TRUNC *d*, где *d* – целое число 0, 1, 2 или 3 – это число десятичных цифр, которое надо сохранить в значении.

По умолчанию/незаданию подкоманды, DISCRET= RND 0, т.е. макрос на входе округляет всякое значение до целого. DISCRET срабатывает *после* просеяния значений подкомандой VALUES.

ПРИМЕР 3.

!KO\_hfreq vars= v1 to v50 /cap= f# /values= 2 3 6 /discret= TRUNC 1.

* Макрос будет подсчитывать частоты только у значений 2, 3, 6 (остальные значения данных будут приравнены к пропускам). Поскольку п/к VALUES срабатывает до подкоманды DISCRET, а значения 2, 3, 6 – целые, без десятичных цифр, то DISCRET= TRUNC 1 в данном случае это то же, что TRUNC 0 или дефолтное RND 0.

**CAP**

Приставка в имена переменных выходящего массива, с частотами значений. Ее можно взять в кавычки/апострофы (они рекомендуются, если приставку вы оканчиваете на точку: “var.”).

**ENDING**

По умолчанию и при ENDING=VALUE имена переменных выходящего массива оканчиваются на собственно значение (если значение отрицательно, то “-“ заменяется на “\_”). Например, если CAP= *F#*, то переменные, соответствующие значениям: *-2.5, -2, 0, 1, 3.2*, будут именоваться: *F#\_2.5, F#\_2, F#0, F#1, F#3.2*. (Вы можете использовать другой символ вместо “\_” в качестве заместителя отрицательного знака; для этого добавьте подкоманду /NEGSIGN= '*symb*', где *symb* это желаемый символ, который разрешено использовать в именах переменных.)

При ENDING=INDEX будут использованы порядковые номера по возрастанию значений. Имена будут тогда *F#1, F#2, F#3, F#4, F#5*.

***Особые режимы***

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

***Родственные макросы***

* !KO\_HFREQ – довольно универсален. Им можно получить фиктивные переменные, какие выдают макросы !KO\_CATBIN, !KO\_!ACATBIN/!KO\_!ACATBIN2 (коллекция “Categorical - Binary recodings”). Если вы введете единственную переменную в !KO\_HFREQ, то на выходе будут переменные с частотой 1 или 0, т.е. двоичные фиктивные переменные. Введение же множественных переменных сымитирует макросы !KO\_MRCMRD, !KO\_!AMRCMRD/!KO\_!AMRCMRD2 (коллекция “Categorical - Binary recodings”).

# МАКРОС !KO\_HCOUNT: ПОДСЧЕТ ПО ГОРИЗОНТАЛИ УНИКАЛЬНЫХ ЗНАЧЕНИЙ И ПОВТОРОВ

Version 2, Jan 2021 (Version 1, Feb 2006). Tested on SPSS Statistics 20, 22, 25.

!KO\_hcount vars= *v1 to v10* /\*Числовые переменные (поименно и/или ч-з to), в которых горизонтально

/\*подсчитывать уникальные значения и повторы

/values= /\*Опционально: среди каких именно значений подсчитывать

/missing= INCLUDE /\*User-missing значения не считать (EXCLUDE, тж п/у) или считать

/\*как валидные (INCLUDE)

/rtimes= /\*Опционально: число различных значений, повторяющихся именно сколько раз, показывать

/cap= *'cnt\_'* /\*Опционально: добавить приставку в имена вычисляемых переменных.

Минимум надо задать VARS.

Подсчитывает, внутри наблюдений, сколько встречается различных значений («уникальных значений»), сколько из них неповторяющихся и повторяющихся, и каково наибольшее и наименьшее число повторов среди значений. Макрос создает (или обновляет, если уже есть в массиве) 5 переменных – *UNIQ#*, *UNIQ\_S#*, *UNIQ\_R#*, *R\_MAX#*, *R\_MIN#*, показывающих эти количества.

Допустим, данные респондента есть: 8 2 3 2 7 3 5 2.

Тут имеем:

* Различных встреченных значений (*UNIQ#*): **5** (именно, 2, 3, 5, 7, 8)
* Из них однократно встречающихся (*UNIQ\_S#*): **3** (именно, 5, 7, 8)
* И встречающихся повторно (*UNIQ\_R#*): **2** (именно, 2, 3)
* Наибольшее наблюдаемое число повторов (*R\_MAX#*): **3** (значение 2 встречено 3 раза)
* Наименьшее наблюдаемое число повторов (*R\_MIN#*): **2** (значение 3 встречено 2 раза)

Макрос выдает пять выделенных значений.

Макрос подсчитывает только валидные значения. Вы можете затребовать считать пользовательские пропущенные значения валидными. Вы можете также задать некоторые другие условия.

В рабочем массиве должно не быть переменных *NVARS###* и *NVARS##$*.

ПРИМЕР 1.

!KO\_hcount vars= v1 to v10.

* Подсчет по горизонтали, сколько различных валидных значений, повторяющихся и неповторяющихся, содержится в V1 – V10.

***Подкоманды***

**VARS**

Числовые переменные, в которых вести горизонтально подсчет. Список поименно и/или ч-з «to». Если переменных очень много, рекомендуется использовать «to»: это может сэкономить время исполнения.

**VALUES**

По умолчанию макрос делает подсчет среди всех встреченных валидных значений. Здесь вы можете указать интересующий вас диапазон значений или их список. Тогда подсчет будет только среди этих встреченных значений. Для указания диапазона используйте следующие ключевые слова (принятые в SPSS-командах COUNT и RECODE): THRU, LO (или LOWEST), HI (или HIGHEST). Например, 0 6 THRU HI будет означать: диапазон от 6 и выше и еще значение 0.

Учтите, что если в VALUES в заданную область попадают пользовательские пропущенные значение (user-missing) и MISSING=EXCLUDE, то эти значения не будут считаться в тех переменных, в каких они user-missing.

ПРИМЕР 2.

!KO\_hcount vars= v1 v4 to v41 /values= -5 THRU 12 13 THRU 19.

!KO\_hcount vars= v1 v4 to v41 /values= 10 12 16 99 /missing= INCLUDE.

* В первой команде «горизонтальные» подсчеты посвящены только (валидным) значениям в диапазоне от -5 до 12 и 13 до 19.
* Во второй команде «горизонтальные» подсчеты посвящены только значениям из списка 10, 12, 16, 99, причем пользовательские пропуски переменных VARS макрос принимает за валидные значения.

**MISSING**

Вы можете считать валидными пользовательские пропущенные значения в VARS при подсчете: MISSING=INCLUDE. По умолчанию/незаданию MISSING=EXCLUDE и пропущенные значения игнорируются.

**RTIMES**

Эта подкоманда затрагивает только переменную *UNIQ\_R#*. По умолчанию *UNIQ\_R#* показывает, сколько различных значений встречаются более раза, неважно сколько именно раз. С помощью RTIMES можно попросить показать, сколько различных значений встречаются конкретно по столько-то раз. Укажите оператор (=, >, <, >=, <=, <>) и число раз. Например, RTIMES= =3 покажет, сколько различных значений встречаются по 3 раза. RTIMES= >3 покажет, сколько различных значений встречаются чаще, чем по 3 раза. RTIMES= <>3 покажет, сколько различных значений повторяются не по 3 раза (т.е. встречаются по 2 или 4 или 5 и т.д. раз).

Вместо числа вы можете указать имя переменной, содержащей нужные целые положительные числа (т.е. для каждого наблюдения можно предусмотреть свое такое число).

RTIMES= =1 является ошибкой, т.к. переменная *UNIQ\_R#* посвящена именно повторяющимся значениям; однократным значениям посвящена *UNIQ\_S#*. RTIMES= >1 эквивалентно незаданию п/к RTIMES.

**CAP**

По умолчанию макрос создает или обновляет в массиве данных переменные с именами *UNIQ#*, *UNIQ\_S#*, *UNIQ\_R#*, *R\_MAX#*, *R\_MIN#* (если у этих переменных есть user-missing дефиниции, макрос сразу снимает их). Подкомандой CAP вы можете затребовать создать новые переменные с этими именами, но с пристакой. Задайте приставку. Ее можно взять в кавычки/апострофы (они рекомендуются, если приставку вы оканчиваете на точку: “var.”). Если вы задаете приставку, то во входящем массиве должно еще не быть переменных с этими составными именами.

***Особые режимы***

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

***Родственные макросы***

* Подсчет конкретных повторяющихся значений, в частности с наложением запрета на учет некоторых ячеек данных, можно выполнить макросом !KO\_VALVARS (“Series Response tools”).

# МАКРОС !KO\_HGAPS: СТАТИСТИКА РАЗРЫВОВ МЕЖДУ ЗНАЧЕНИЯМИ В ВАРИАЦИОННОМ РЯДУ ПО ГОРИЗОНТАЛИ

Version 2, Jan 2021 (Version 1, Nov 2007). Tested on SPSS Statistics 20, 22, 25.

!KO\_hgaps vars= v1 to v10 /\*Числовые переменные (поименно и/или ч-з to), среди которых

/\*вычислять по-горизонтали разрывы

/sorted= /\*Значения в наблюдениях сортированы (YES) или нет (NO, тж п/у)

/missing= INCLUDE /\*User-missing значения не считать (EXCLUDE, тж п/у) или считать

/\*как валидные (INCLUDE)

/cap= /\*Опционально: добавить приставку в имена вычисляемых переменных.

Минимум надо задать VARS.

Разрыв это абсолютная разница, расстояние между двумя смежными неодинаковыми значениями в вариационном ряду данных, т.е. в ряду, сортированном по возрастанию (или убыванию) значений. Для каждого наблюдения массива данных макрос вычислит статистику разрывов: число существующих в нем разрывов, минимальный, максимальный разрывы, усредненный разрыв, стандартное отклонение разрывов, создав соответственно пять переменных: *NGAP#*, *MINGAP#*, *MAXGAP#*, *MEANGAP#*, *SDGAP#*. Статистика разрывов полезна, в частности, при изучении выбросов и степени неоднородности в данных.

Допустим, данные респондента есть: 9 0 3 2 8 3 5 11.

Тогда, как легко убедиться если расположить данные по величине: 0 2 3 3 5 8 9 11,

* Всего имеем *NGAP#* = 6 разрывов: (0-2) (2-3) (3-5) (5-8) (8-9) (9-11)
* Наименьший разрыв, *MINGAP#*, между (неодинаковыми) значениями равен **1**: именно, 3-2=1, также 9-8=1
* А наибольший разрыв, *MAXGAP#*, равен **3**: именно, 8-5=3
* Средняя величина всех разрывов, *MEANGAP#*, равна [(2-0)+(3-2)+(5-3 )+(8-5)+(9-8)+(11-9)] / 6 = **1.83**
* И стандартное отклонение их, *SDGAP#*, равно **0.75**.

Макрос выдаст пять выделенных значений. Число разрывов *NGAP#* всегда равно числу различных значений минус 1. Если все значения у наблюдения одинаковы, во всех пяти переменных будет 0. Если все значения у наблюдения есть пропуски, в *NGAP#* будет - 1, а в остальных переменных будет 0.

Макрос обрабатывает только валидные значения. Вы можете затребовать считать пользовательские пропущенные значения валидными.

В рабочем массиве должно не быть переменных *NVARS###* и *NVARS##$*.

***Подкоманды***

**VARS**

Числовые переменные, среди которых анализировать горизонтально разрывы. Список поименно и/или ч-з «to». Если переменных очень много, рекомендуется использовать «to»: это может сэкономить время исполнения. Значения не обязаны быть сортированными слева направо. Данные принимаются за количественные.

**SORTED**

По умолчанию и при SORTED=NO макрос использует алгоритм, не требующий сортированности данных; однако он медлен на больших данных (т.е. многих наблюдений и большого числа переменных VARS). Если в ваших переменных VARS (валидные) значения сортированы внутри наблюдений, то вы можете указать SORTED=YES; тогда макрос использует более быстрый алгоритм. Не важно, по возрастающей или по убывающей имеется сортированность, и перемежаются ли валидные значения пропусками или нет. Вы можете предварительно отсортировать значения в наблюдениях макросом !KO\_HSORT или (быстрее) макросом !KO\_HQSORT.

**MISSING**

Вы можете считать валидными пользовательские пропущенные значения в VARS: MISSING=INCLUDE. По умолчанию/незаданию MISSING=EXCLUDE и пропущенные значения игнорируются.

**CAP**

По умолчанию макрос создает или обновляет в массиве данных переменные с именами *NGAP#*, *MINGAP#*, *MAXGAP#*, *MEANGAP#*, *SDGAP#* (если у этих переменных есть user-missing дефиниции, макрос сразу снимает их). Подкомандой CAP вы можете затребовать создать новые переменные с этими именами, но с пристакой. Задайте приставку. Ее можно взять в кавычки/апострофы (они рекомендуются, если приставку вы оканчиваете на точку: “var.”). Если вы задаете приставку, то во входящем массиве должно еще не быть переменных с этими составными именами.

***Особые режимы***

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

1. Управление зерном случайных чисел в SPSS Statistics: меню Transform – Random Number Generator. [↑](#footnote-ref-1)