***Impute missing data***

SPSS macros by Kirill Orlov

kior@akado.ru, ttnphns@gmail.com

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

All rights reserved.

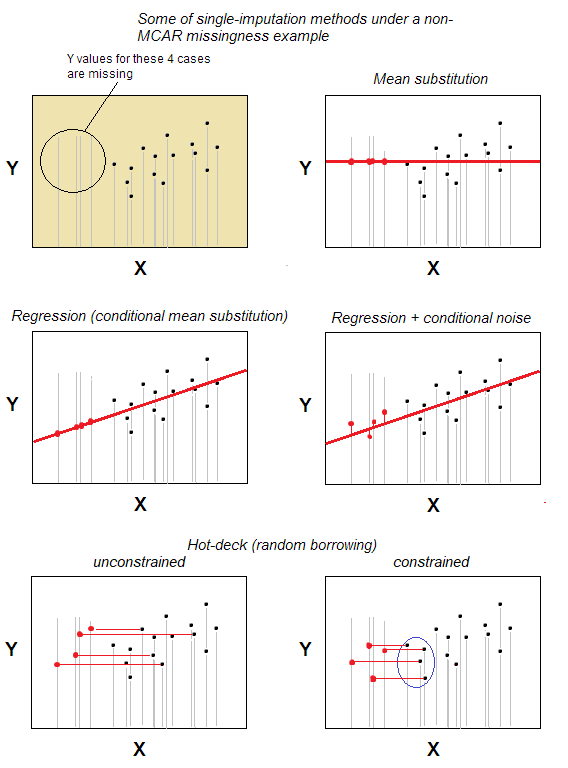
*Импутация пропущенных данных*. Макросы выполняют колодное замещение (hot-deck imputation) пропущенных значений, заимствуя валидные значения у наблюдений, которые похожи на наблюдения с пропусками по неким фоновым характеристикам. Отдельный макрос выполняет произвольное, заданное пользователем заимствование значений у одних наблюдений другими наблюдениями.

*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.

* [!KO\_HDIMPUT](#_МАКРОС_!HDIMPUTE:_КОЛОДНАЯ_1)E и [!KO\_HOTDECK](#_МАКРОС_!HOTDECK:_КОЛОДНАЯ) исполняют одно и то же задание – колодную импутацию, но эти макросы различаются опциями и алгоритмами, поэтому результат импутации (даже под одним и тем же зерном случайных чисел) там и там будет обычно разный. Метод колодной (hot-deck) импутации состоит в том, что наблюдение с пропущенным значением получает валидное значение от какого-либо случайно выбранного наблюдения из тех, что достаточно похожи на него по заданным пользователем *фоновым* переменным (которые еще называют «колодой», deck-переменными). Колодная импутация популярна, поскольку она и проста в идее и одновременно приемлема для ситуаций, когда такие простые методы обработки пропущенных значений как списочное их исключение (listwise deletion) или замена на среднее/медиану (mean/median substitution) не годятся по причине того, что пропуски размещены в данных не хаотично, не по паттерну MCAR (Missing Completely At Random). Упомянутые фоновые переменные должны быть категориальные. Заметим вообще, что любую импутацию пропущенных данных рекомендуют делать, только если пропусков не более чем примерно 20% от числа наблюдений в переменной.
* [!KO\_BORROW](#_МАКРОС_!BORROW:_ЗАИМСТВОВАНИЕ) предназначен для ручной, задаваемой пользователем перекидки значений (всяких, не обязательно только валидных, в общем случае) с наблюдения на наблюдение в пределах переменной («вертикально»). Этот макрос можно употребить в частности для того, чтобы распространить импутационное решение с одной на множество переменных.

**Рис. 1. Колодная импутация в ряду некоторых других методов единичной импутации**. Колодная импутация не моделирует и не оценивает значения для пропусков, а выбирает их случайно из массы наличных валидных наблюдений, если надо – под ограничительными условиями.



# МАКРОС !KO\_HDIMPUTE: КОЛОДНАЯ (HOT-DECK) ИМПУТАЦИЯ (ТЩАТЕЛЬНЫЙ АЛГОРИТМ)

Version 3, Jun 2020 (Version 1, Sep 2011). Tested on SPSS Statistics 20, 22, 25.

*Этот макрос требует SPSS Statistics 17 или выше.*

!KO\_hdimpute vars= *income nchildren* /\*Поименно числовые переменные с пропусками

/nmiss= 40 36 /\*Сколько пропусков пытаться заместить в каждой:

/\*целые числа по числу VARS или *число* ALL

/bgvars= *marriage position agegr* /\*Поименно категориальные фоновые переменные,

/\*по которым сопоставлять

/match= /\*Тип сопоставления: искать полное совпадение (ALL, тж п/у),

/\*частичное совпадение (ANY *число*), иерархическое частичное совпадение (HIE *число*)

/varsbg= *income* /\*Учитывать корреляции среди этих переменных (поименный список переменных

/\*из числа VARS)

/discret= /\*Если задано VARSBG: мерные переменные в том списке

/mult= ATONE /\*Для множественных VARS: стараться импутировать все пропуски реципиента "заодно",

/\*одним донором (ATONE, тж п/у), или импутировать их независимо (INDEP)

/samp= NOREPLACE /\*Использовать донора: строго однократно (NOREPLACE, тж п/у),

/\*или не более стольких-то раз (число)

/\*или разрешить повторное использование без ограничений (REPLACE)

/print= YES /\*Динамическая распечатка процесса импутации: YES или NO (тж п/у).

Минимум надо задать VARS, NMISS, BGVARS.

Макрос осуществляет над переменными с пропусками (импутационными переменными) операцию колодной импутации. Каждое наблюдение с пропуском называется реципиентом. Наблюдения, похожие на реципиента и имеющие в этой же переменной валидные значения, являются потенциальными донорами; среди них один выбирается случайно и становится донором валидного значения для реципиента. Если потенциальных доноров нет, реципиент остается с пропуском (неудавшаяся импутация).

Сколько в каждой импутационной переменной пытаться заместить пропусков – вы должны в этом макросе указать. Вы можете выбрать режим, по которому искать наблюдения, похожие на реципиента фоновыми характеристиками. Вы можете опционально учитывать ассоциации между переменными с пропусками. Наконец, можно контролировать повторное использование наблюдения в качестве донора.

Для каждой переменной *ИМЯ* из VARS макрос создает в рабочем массиве данных: (1) переменную IMPUT\_*ИМЯ* с импутированными значениями вдобавок к исходно валидным; ее можно использовать в дальнейшем анализе; (2) переменную DONOR\_*ИМЯ*, показывающая, для каждого наблюдения-пропуска в массиве, его финальное состояние: либо номер наблюдения-донора, импутировавшего сюда, либо пометку, что импутация не случилась по причине ненахождения алгоритмом донора.

Макрос создает временные переменные с именами, оканчивающимися на *#$#$*. Рекомендуется поэтому, чтобы во входящем массиве данных не было переменных с именами, имеющими такое окончание.

Результат зависит от генерации случайных чисел и меняется от пуска к пуску. Регулировать зерно случайных чисел (например чтобы повторить результат) вы можете в меню SPSS Transform – Random Number Generators.

*Быстродействие*. Макрос не является быстрым в исполнении, т.к. очередно подыскивает донора к каждому наблюдению с пропусками или даже каждому пропущенному значению. По этой причине, для больших массивов, многих переменных VARS или большого числа пропусков NMISS более скорый [!KO\_HOTDECK](#_МАКРОС_!HOTDECK:_КОЛОДНАЯ) может быть рекомендован, хотя он менее гибок и тщателен. А также см. ПРИМЕР 3. Сравнение двух макросов:

|  |  |
| --- | --- |
| !KO\_HDIMPUTE | !KO\_HOTDECK |
| Медленнее | Быстрый (рекомендован для большого массива, многих переменных или многих пропусков) |
| Импутирует указанное число пропусков | Импутирует все пропуски |
| Если есть донор, то импутация гарантирована | Если есть донор, то импутация очень вероятна |
| Разрешает частичное совпадение по набору фоновых переменных | Нет |
| Донор выбирается из подходящих кандидатов случайно; в режиме частичного совпадения вероятность выбора зависит от силы (потенциальности) кандидата | Донор выбирается из подходящих кандидатов случайно (они все равнопотенциальны) |
| Может учитывать ассоциации между импутационными переменными (constrained hot-deck) | Нет |
| Есть режим «заодно» и «независимо» | Смешанный режим |
| Есть запрет повторного использования донора. Можно поставить предел числу раз использования | Есть ослабление вероятности повторного использования донора |
| Порядок в VARS может иметь значение | Порядок в VARS почти не играет роли |
| Поддерживает фильтрованность массива | Нет |

**Алгоритм**

Принцип: рассматривать каждый индивидуальный пропуск и подыскивать ему донора. !KO\_HDIMPUTE обрабатывает переменные VARS очередно, и внутри каждой из них наблюдения с пропусками (*реципиенты*) обрабатываются очередно.

1) Набор реципиентов. При обработке очередной переменной наблюдения-пропуски отделяются от валидных наблюдений и сортируются в случайном порядке, и NMISS первых из них назначаются пройти импутацию (быть реципиентами). Причем те наблюдения-пропуски, которые уже обрабатывались (были реципиентами) при предшествующих переменных, ставятся во главу этой случайной очереди NMISS реципиентов – чтобы импутироваться прежде остальных: так соблюдается преемственность между VARS в отношении набора реципиентов.

2) Выбор донора. Обрабатываемому реципиенту (в очереди NMISS таковых) подыскиваются кандидаты – *потенциальные* доноры. Это валидные по текущей переменной наблюдения, похожесть которых на реципиент по фоновым переменным BGVARS не ниже порога, задаваемого условием MATCH. Сама же величина похожести, превысившая порог, может быть у потенциальных доноров индивидуальной и называется их силой, или *потенциальностью P*. Потенциальный донор с максимальным значением *P2U* (где *U* это случайное число из равномерного распределения, порожденное немедленно и для каждого потенциального донора независимо) реализуется как *донор* и импутирует свое значение в реципиент. Другими словами, донор выбирается случайно, но вероятность выбора пропорциональна квадрату потенциальности. См. *Примечение*.

3) Импутация. При MULT=ATONE донор «заодно» импутирует и те последующие VARS реципиента, которые есть пропуски и куда донор может предоставить валидное значение (т.е. валиден по этим VARS). При MULT=INDEP донор ограничивается импутацией только в текущей переменной, так что импутация разных VARS становится независимой (будет делаться вероятно разными донорами для одного и того же наблюдения-реципиента).

Импутированный валидным значением реципиент не может стать донором этого импутированного значения (лавинная импутация запрещена). Если реципиент был исходно валиден в другой, чем текущая, переменной VARS, он может стать донором, когда обрабатываться та переменная.

*Примечание*: При задании п/к VARSBG (см.) сами переменные VARS в своем исходном виде рассматриваются как еще один набор фоновых переменных, всегда с пороговым условием похожести, равносильным “MATCH= ANY 1”. Тогда потенциальными донорами будут считаться только те валидные наблюдения, которые удовлетворяют *обоим* пороговым условиям: этому и тому, что было указано в п/к MATCH. Что касается величины потенциальности *P* у потенциальных доноров, то она есть *сумма* двух похожестей: согласно тому и согласно другому условию. Например, пусть VARS= V1 V2 V3, BGVARS= G1 G2 G3, MATCH= ANY 2, VARSBG= V1 V2 V3. Тогда, чтобы попасть в потенциальные доноры, валидное наблюдение должно не только совпадать с реципиентом по каким-то двум из G1, G2, G3, но и быть похожим на реципиент хотя бы по одному из V1, V2 или V3 (что логически требует уже, чтобы реципиент был валиден хотя бы в одной из этих VARSBG-переменных). Пускай реципиент есть пропуск в V1, но валиден в V2 и V3, и некий потенциальный донор совпадает с ним только по V3, и также по всем G1, G2, G3. Тогда сила *P* данного потенциального донора = 1+3=4.

***Подкоманды***

**VARS**

Поименный список числовых переменных с пропусками, которые подвергнуть импутации. Пропуски могут быть как системными (system-missing), так и пользовательскими (user-missing); замещаться будут и те и другие. Указание в списке переменных, в которых пропусков нет, не ведет к ошибке, так что допустимо.

Порядок переменных в списке может иметь значение, особенно в условиях /SAMP=NOREPLACE и /MULT=INDEP при ограниченном пуле потенциальных доноров и достаточно длинном списке VARS. Поскольку пропуски импутируются сначала все в одной переменной, потом уже в другой, то переменные, стоящие в хвосте VARS, могут в принципе испытать нехватку доноров вообще или нехватку более «сильных» (более потенциальных) доноров.

**NMISS**

Укажите, сколько пропусков вы хотите заместить валидными значениями в каждой переменной VARS. Укажите свое число для каждой переменной (тогда NMISS будет содержать столько чисел, сколько переменных VARS), или укажите одно число и после него кл. слово ALL (это число будет распространено на все VARS). Числа должны быть целыми неотрицательными.

Максимальное допустимое число – 1000. Оно совпадает с пределом числа макроциклов в SPSS Statistics. Однако вы сможете задать большее число, если поднимете указанный предел командой SET MITERATE (см. *Command Syntax Reference*, SET command). В любом случае, если вы укажете в NMISS число больше, чем установленный предел числа макроциклов, SPSS Statistic не исполнит макрос и сообщит об ошибке.

Если заданное для переменной число *n* меньше, чем реально есть пропусков в переменной, тогда для процедуры импутации будет случайно отобрано *n* пропусков, а остальные пропуски в переменной будут проигнорированы. Если вы зададите *n* больше, чем есть пропусков в переменной, это не испортит результат, но макрос будет работать дольше из-за холостых оборотов.

Существует преемственность среди VARS-переменных: наблюдения, импутированные в прошлых переменных, импутируются прежде остальных при обработке следующей переменной.

Вы вправе использовать фильтр для отбора наблюдений для импутации (см. «Особые режимы»).

**BGVARS**

Укажите поименный список фоновых переменных, по которым сопоставлять реципиента с остальными наблюдениями в поиске потенциальных доноров. Переменные должны быть числовые. Они должны быть категориальными; причем от количества категорий зависит успех поиска потенциальных доноров: чем меньше категорий, тем легче найти потенциальных доноров. Континуальную переменную дискретизируйте в категориальную. Если фоновых переменных у вас нет (что значит, что всякое валидное наблюдение массива может быть для вас потенциальным донором), просто создайте в массиве числовую переменную-константу с валидным значением и укажите ее.

Только валидные значения BGVARS-переменных могут стать основанием для сродства реципиента и донора.

В списке не могут присутствовать переменные из списка VARS (см. для них отдельную подкоманду VARSBG).

**MATCH**

Эта подкоманда играет роль, когда фоновых переменных несколько, и задает режим сопоставления наблюдений по фоновым переменным. Совпадение только валидными значениями идет в счет. Выберите:

ALL - (тж. по умолчанию/незаданию) Чтобы валидное наблюдение попало в потенциальные доноры, оно должно совпадать с реципиентом по всем фоновым переменным: это режим полного совпадения.

ANY *n* - Чтобы валидное наблюдение попало в потенциальные доноры, оно должно совпасть с реципиентом по любым в списке BGVARS *n* переменным. Чем по большему числу любых в списке BGVARS переменных есть совпадение, тем потенциальнее становится данное наблюдение как потенциальный донор.

HIE *n* - Чтобы валидное наблюдение попало в потенциальные доноры, оно должно совпасть с реципиентом по первым в списке BGVARS *n* переменным. Чем по большему числу первых в списке BGVARS переменных есть совпадение, тем потенциальнее становится данное наблюдение как потенциальный донор. Этот вариант чувствителен к порядку переменных в списке BGVARS; более важные для сопоставления переменные пишите раньше.

Число *n* должно быть целым числом от 1 до число\_фоновых\_переменных-1.

Подбор по фоновым характеристикам, заказанный в MATCH, дает коллекцию потенциальных наблюдений-доноров для текущего реципиента. Если список потенциальных доноров окажется пуст, импутации реципиенту не будет. Если потенциальных доноров более одного, выбор донора из них для реципиента делается случайно. В режимах частичного совпадения – MATCH= ANY *n* или HIE *n* – набранные потенциальные доноры могут различаться своей потенциальностью *P* (степенью совпадения с реципиентом) и случайный выбор между ними взвешивается квадратом потенциальности: донором становится наблюдение с максимальной величиной *P2 \* случайное число*, или, др. словами, *P* \* rv[~U(0*, P*)], где U есть равномерное распределение.

ПРИМЕР 1.

!KO\_hdimpute vars= v1 /nmiss= 120 /bgvars= region sex agegr marriage /match= HIE 3.

* Импутация пропусков в *V1* с контролем региона проживания респондента, его пола, возрастной группы, брачного статуса. Чтобы валидное наблюдение имело шанс быть донором, оно должно совпасть с наблюдением-пропуском по меньшей мере по трем первым из четырех этих переменных. Если совпадает по всем четырем – шанс стать донором увеличивается.
* Импутации подвергнутся 120 случайно выбранных пропусков. Или столько, сколько есть их в *V1*, если их там меньше, чем 120.

**VARSBG, DISCRET**

Эти подкомандвы действуют, если в VARS более одной переменной. Подкоманда VARSBG позволяет сделать “constrained” колодную импутацию, т.е. учитывающую (до-импутационные) ассоциации/корреляции между самими переменными VARS (см. рисунок в начале). Укажите в VARSBG поименно те переменные из VARS, связи между которыми и каждой импутируемой переменной из VARS вы хотите учитывать во время импутации последней. Вы можете указать в VARSBG одну, несколько или все переменные списка VARS.

Технически проблема учета ассоциаций между импутируемыми переменными решается тем, что список VARSBG образует еще один, вдобавок к BGVARS, набор фоновых переменных. Это ужесточает требования к потенциальным донорам: донор должен не только совпадать с реципиентом по BGVARS (по условию MATCH), но и совпасть по (как минимум одному из) VARSBG. Совпадение по VARSBG и есть защита ассоциаций/корреляций среди VARS от ее «размыва» импутационным процессом. Когда наблюдение-реципиент есть пропуск в переменной X из VARS, но не пропуск в каких-либо переменнных Y из VARSBG, то ему подбирается донор, который совпадает с ним по этим переменным Y. Донор обязан совпасть как минимум по одному из Y, но потенциальность донора тем выше, чем по большему числу Y он совпадет с реципиентом (см. выше «Алгоритм, *Примечание*»). Если реципиент есть пропуск во всех VARSBG, то он останется неимпутирован, т.к. неясно, какого донора ему искать (вы можете доимпутировать в такие реципиенты повторным пуском макроса, уже без задания VARSBG).

Подкоманда DISCRET является подсобной для подкоманды VARSBG (см.). Если в VARSBG имеются некатегориальные переменные – континуальные, мерные, то вы должны перечислить их в DISCRET – чтобы макрос дискретизировал их (внутренне, не изменяя данные в массиве) в категориальные. Это необходимо, потому что фоновые переменные, какими являются VARSBG, должны быть категориальными. Для категоризации макрос использует SPSS-команду RANK /NTILES(*k*). Он разбивает диапазон количественной переменной, имеющей > *k* уникальных значений, на *k* примерно равных числом наблюдений групп. По умолчанию *k*=5, но вы можете изменить этот параметр, добавив макрокоманде подкоманду /KBINS= *число*.

ПРИМЕР 2.

!KO\_hdimpute vars= v1 v2 v3 /nmiss= 45 76 61 /bgvars= region sex /varsbg= v1 v2 v3 /discret= v1 v2.

* Импутация с контролем региона проживания и пола респондента, а также с намерением учесть, по возможности, ассоциации между самими импутируемыми переменными VARS: они указаны в п/к VARSBG. Причем переменные *V1* и *V2* из них не категориальные, поэтому их требуется дискретизировать (п/к DISCRET).

**MULT**

Подкоманда действует, если переменных VARS несколько. Она устанавливает манеру импутации в наблюдение, которое есть пропуск в более чем одной переменной VARS.

ATONE - (тж по умолчанию) импутация «заодно». Наблюдение-донор импутирует пропуск наблюдения-реципиента в текущей (обрабатываемой) переменной и в последующих в списке VARS переменных, для которых реципиент есть пропуск, а донор есть валид.

INDEP - импутация «независимая». Наблюдение-донор импутирует пропуск наблюдения-реципиента только в текущей переменной.

MULT=ATONE означает, что реципиент с множественными пропусками скорее всего будет замещен каким-то одним донором (что можно понять как создание наблюдения-дупликата из донора), а при MULT=INDEP такой реципиент вероятно будет импутирован несколькими разными донорами (что можно понять как создание наблюдения-гибрида доноров). Если дуплицирование наблюдений в некотором массиве обычно сохраняет корреляционную конфигурацию, то гибридизация наблюдений в массиве с выраженными корреляциями обычно имеет тенденцию ослаблять их.

Поддержка корреляций между VARS зависит и от специальной п/к VARSBG. Если MULT=ATONE стремится создать дупликат донора на месте реципиента с множеством пропусков, то VARSBG использует непропуски реципиента, если такие есть, для подбора ему корреляц-сберегающего донора.

ПРИМЕР 3. Ускорение работы с помощью макроса [!KO\_BORROW](#_МАКРОС_!BORROW:_ЗАИМСТВОВАНИЕ) в ситуации, когда наблюдения-пропуски - сквозные.

!KO\_hdimpute vars= v1 v2 v3 v4 v5 v6 v7 v8 /nmiss= 100 ALL /bgvars= region sex /mult= ATONE.

* Пропуски в переменных *V1 – V8* по большей части принадлежат одним и тем же наблюдениям, т.е. они преимущественно сквозные. Макрос симпутирует в режиме MULT=ATONE (тж. по умолчанию), т.е. все пропуски наблюдения-реципиента он будет стараться, по возможности, заместить одним наблюдением-донором.
* Восемь переменных VARS это много: макрос будет готовиться и исполняться не быстро. Альтернативный скорый подход в этой ситуации, дающий достаточно сходный результат – импутировать только одну переменную, а затем распространить ее результаты на остальные переменные макросом !KO\_BORROW.

!KO\_hdimpute vars= v1 /nmiss= 100 /bgvars= region sex.

!KO\_borrow vars= v2 v3 v4 v5 v6 v7 v8 /donor= donor\_v1 /cap= 'impute\_'.

* Только одна переменная, *V1*, обрабатывается в !KO\_HDIMPUTE, что – быстро. На выходе будет переменная *IMPUTE\_V1* и переменная *DONOR\_V1*; последняя содержит номера «сыгравших» наблюдений доноров.
* !KO\_BORROW, пользуясь этой переменной в подкоманде DONOR, заставляет указанные доноры дать свои валидные значения в переменные *V2 – V8*. Поскольку пропуски в этих переменных это по-большей части те же наблюдения, что были пропущены в *V1*, то результат такого действия близок к тому, что было в первоначальной !KO\_HDIMPUTE со всеми 8 переменными.

**SAMP**

Эта подкоманда заведует повторным использованием наблюдений в качестве доноров. По умолчанию/незаданию и при SAMP=NOREPLACE наблюдение-донор может импутировать только в одно наблюдение-реципиент; другими словами, выбор донора это «выбор без возвращения» его в коллектив потенциальных доноров. При SAMP=REPLACE наблюдение, ставшее донором для наблюдения-реципиента, сохраняет шанс импутировать и в другие наблюдения-реципиенты; другими словами, выбор донора это «выбор с возвращением». Вы можете также указать SAMP=*число* (укажите целое положительное), и тогда наблюдение не сможет импутировать больше, чем в это число наблюдений-реципиентов. SAMP=1 эквивалентно SAMP=NOREPLACE.

**PRINT**

Эта подкоманда позволяет выводить в окно результатов отчет о процессе импутации, во время исполнения. Это удобно, чтобы следить за ее ходом и порядком, когда пропусков заказано импутировать много. Укажите PRINT=YES. По умолчанию PRINT=NO и вывода не будет.

***Особые режимы***

Макрос не слушается взвешивания. Он не рассчитан на расщепленное состояние массива данных (SPLIT FILE) и на отбор наблюдений командой USE. Не пускайте перед макросом временные операции (под командой TEMPORARY). Макрос слушается фильтрации (FILTER), которой можно управлять отбором наблюдений для участия в импутации. Выфильтрованное валидное наблюдение не выступит донором, а выфильтрованное наблюдение-пропуск получит на выходе статус «не подвергалось импутации».

***Некоторые вопросы***

*В чем разница между подкомандами BGVARS и VARSBG?* Обе подкоманды задают списки фоновых переменных (для поиска донора, подходящего реципиенту). Разница между ними не теоретическия, а удобственная. BGVARS это обязательная подкоманда, где надо указать фоновые переменные, не входящие в список VARS. Предполагается, что пропусков в BGVARS-переменных нет или мало, так что сами они не нуждаются в импутации. VARSBG это опциональная подкоманда, которой вы можете импутационные VARS-переменные привлечь и на роль фоновых. Условие совпадения по BGVARS-переменным задается вами в п/к MATCH (по умолчанию оно есть “ALL”). Условие совпадения по VARSBG-переменным предзадано макросом и всегда эквивалентно “ANY 1” (такое либеральное условие объясняется тем, что во входящих VARS-переменных может быть много пропусков). BGVARS-переменные должны быть уже категориальными на входе. VARSBG-переменные, если они не категориальные, надо указать в DISCRET, чтобы макрос внутренне сначала скатегоризовал их.

*В чем разница между подкомандами SAMP и MULT?* SAMP это о том, в одно или в множество *наблюдений с пропусками* разрешено импутировать одному наблюдению-донору. MULT это о том, одно или множество *пропущенных значений* у наблюдения-реципиента разрешено импутировать наблюдению-донору. MULT играет роль, если у реципиента может быть несколько пропусков – т.е. если VARS более одной переменной.

# МАКРОС !KO\_HOTDECK: КОЛОДНАЯ (HOT-DECK) ИМПУТАЦИЯ (БЫСТРЫЙ АЛГОРИТМ)

Version 3, Jun 2020 (Version 1, Sep 2011). Tested on SPSS Statistics 20, 22, 25.

*Этот макрос написан Терезой Майрс (Teresa Myers, teresaannmyers@gmail.com) и позже модифицирован/улучшен мной.*

!KO\_hotdeck vars= *income nchildren* /\*Поименно числовые переменные с пропусками

/bgvars= *marriage position agegr* /\*Поименно категориальные фоновые переменные,

/\*по которым сопоставлять

/samp= REPLACE /\*Использовать донора: стараться однократно (RESTRREPLACE, тж п/у),

/\*или разрешить повторное использование (REPLACE).

Минимум надо задать VARS, BGVARS.

Макрос осуществляет, как и макрос [!KO\_HDIMPUTE](#_МАКРОС_!HDIMPUTE:_КОЛОДНАЯ_1), над переменными с пропусками операцию колодной импутации. Используется другой алгоритм (техническое решение), чем в !KO\_HDIMPUTE, поэтому результаты, будучи столь же валидные, получаются здесь другие.

Для каждой переменной *ИМЯ* из VARS макрос создает в рабочем массиве данных: (1) переменную IMPUT\_*ИМЯ* с импутированными значениями вдобавок к исходно валидным; ее можно использовать в дальнейшем анализе; (2) переменную DONOR\_*ИМЯ*, показывающая, для каждого наблюдения-пропуска в массиве, его финальное состояние: либо номер наблюдения-донора, импутировавшего сюда, либо пометку, что импутация не случилась по причине ненахождения алгоритмом донора.

Макрос создает временные переменные с именами, оканчивающимися на *#$#$*. Рекомендуется поэтому, чтобы во входящем массиве данных не было переменных с именами, имеющими такое окончание.

Результат зависит от генерации случайных чисел и меняется от пуска к пуску. Регулировать зерно случайных чисел (например чтобы повторить результат) вы можете в меню SPSS Transform – Random Number Generators.

!KO\_HOTDECK гораздо быстрее чем !KO\_HDIMPUTE и легко справляется с большими массивами данных, содержащими много наблюдений с пропусками. !KO\_HOTDECK не имеет ограничений на число импутируемых пропусков: он всегда пытается импутировать все пропуски, сколько их есть. С другой стороны, !KO\_HOTDECK поддерживает только режим исчерпывающего сопоставления по фоновым переменным (это соответствует спецификации /MATCH=ALL макроса !KO\_HDIMPUTE), что делает его менее гибким, чем !KO\_HDIMPUTE. Алгоритм в !KO\_HOTDECK не абсолютно тщательный, поэтому в редких случаях может оставить пропуск неимпутированным и при наличии подходящего донора. См. сравнительную таблицу двух макросов в описании !KO\_HDIMPUTE.

**Алгоритм**

Принцип !KO\_HOTDECK: собрать вместе похожие наблюдения, валидные и пропуски, и затеять передачу значений среди них. (1) Наблюдения массива сортируются по переменным BGVARS. В следствие, тождественные по всем BGVARS наблюдения образуют группы соседствующих наблюдений; причем они идут там в случайном порядке. Внутри группы *реципиенты* (наблюдения с пропусками в VARS) перемешаны с валидными наблюдениями (*потенциальные* доноры). (2) Пробегая наблюдения в каждой такой группе, макрос делает следующее для каждой переменной VARS одной за другой: (3) если наблюдение есть пропуск (реципиент) в переменной, то макрос осматривает его соседние наблюдения (до 3-х позиций ниже и выше его) и берет донором первое встреченное валидное из них, которое и импутирует значение в реципиент. Поскольку последовательность осмотра соседей всегда одна и та же, то реципиент, являющийся пропуском в нескольких VARS, получит импутации в этих VARS одним и тем же или разными донорскими наблюдениями в зависимости от того, как насыщены валидными значениями соседи реципиента. Если наблюдение стало донором реципиенту в первой переменной и оно валидно в остальных VARS, то как правило (хотя не всегда) оно же будет ему донором к тех других VARS.

Импутированный валидным значением реципиент не может стать донором этого импутированного значения (лавинная импутация запрещена).

ПРИМЕР 1.

!KO\_hotdeck vars= v1 v2 /bgvars= region sex agegr marriage.

* Импутация пропусков в *V1* и *V2* с контролем региона проживания респондента, его пола, возрастной группы, брачного статуса. Чтобы валидное наблюдение имело шанс быть донором, оно должно совпасть с наблюдением-пропуском по всем четырем этим переменных.

***Подкоманды***

**VARS**

Поименный список числовых переменных с пропусками, которые подвергнуть импутации. Пропуски могут быть как системными (system-missing), так и пользовательскими (user-missing); замещаться будут и те и другие.

**BGVARS**

Укажите поименный список фоновых переменных, по которым сопоставлять реципиента с остальными наблюдениями в поиске потенциальных доноров. Переменные должны быть числовые. Они должны быть категориальными; причем от количества категорий зависит успех поиска потенциальных доноров: чем меньше категорий, тем легче найти потенциальных доноров. Континуальную переменную дискретизируйте в категориальную. Если фоновых переменных у вас нет (что значит, что всякое валидное наблюдение массива может быть для вас потенциальным донором), просто создайте в массиве числовую переменную-константу с валидным значением и укажите ее.

В списке не могут присутствовать переменные из списка VARS. Если в каких-либо BGVARS есть пропуски, то эти наблюдения не смогут стать донорами, потому что !KO\_HOTDECK приемлет только полное тождество по всем BGVARS между реципиентом и донором, а пропущенное значение не может быть основанием тождества.

**SAMP**

Эта подкоманда заведует повторным использованием наблюдений в качестве доноров. По умолчанию/незаданию и при SAMP=RESTRREPLACE повторное использование наблюдений в качестве доноров сдержано: редко, когда наблюдение-донор сможет импутировать в более чем одно наблюдение-реципиент (и если это происходит, то не в одной и той же переменной VARS). При SAMP=REPLACE наблюдение, ставшее донором для наблюдения-реципиента, сохраняет значительный шанс импутировать и в другие наблюдения-реципиенты.

***Особые режимы***

Макрос не слушается взвешивания. Он не рассчитан на расщепленное состояние массива данных (SPLIT FILE) и на отбор наблюдений командой USE. Не пускайте перед макросом временные операции (под командой TEMPORARY). Макрос не слушается фильтрации (FILTER), в отличие от макроса !KO\_HDIMPUTE.

# МАКРОС !KO\_BORROW: ЗАИМСТВОВАНИЕ ЗНАЧЕНИЙ У ЗАДАННЫХ ДРУГИХ НАБЛЮДЕНИЙ

Version 1, Dec 2012. Tested on SPSS Statistics 20, 22, 25.

*Этот макрос требует SPSS Statistics 17 или выше.*

!KO\_borrow vars= *v1 v2 v3* /\*Переменные, внутри к-рых делать перекидку значений, поименно

/donor= *donor* /\*Переменная с номерами наблюдений-доноров, с каких заимствовать значения

/ifrecval= ANY /\*Заместить значение, если оно: любое (ANY), валидное (VALID),

/\*пропуск (MISSING, тж п/у), системный пропуск (SYSMIS), пустой текст (BLANK)

/ifdonval= VALID /\*Заместить значением, если оно: любое (ANY), валидное (VALID, тж п/у),

/\*пропуск (MISSING), системный пропуск (SYSMIS), пустой текст (BLANK)

/cap= '*new*\_' /\*Опционально: приставка для создания новых переменных.

Минимум надо задать VARS, DONOR.

Данный макрос осуществляет заимствование значений между наблюдениями: перенос значений с наблюдения на наблюдение в пределах той же переменной. Откуда (с какого наблюдения) должно перенять свое значение текущее наблюдение, должно быть записано в особой переменной, имя которой вы укажете в п/к DONOR.

Если у текущего наблюдения (реципиента) указано в переменной DONOR целое положительное число *i* (не большее, чем число наблюдений в массиве), то реципиент позаимствует значение у *i*-того наблюдения (донора), если относительно этих двух значений ­– своего и донорского – выполняются оба условия, IFRECVAL и IFDONVAL, соответственно. Макрос переносит значения «по вертикали», в пределах одной обрабатываемой переменной. Вы можете обработать за один пуск одну или несколько таких переменных.

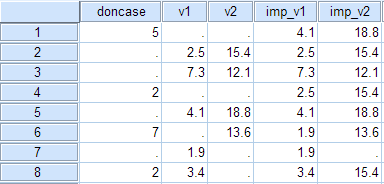
В частности, данный макрос годится для распространения итогов колодной импутации пропусков с одной переменной на множество переменных – с *одних и тех же* наблюдений-доноров. [!KO\_HDIMPUTE](#_МАКРОС_!HDIMPUTE:_КОЛОДНАЯ_1) и [!KO\_HOTDECK](#_МАКРОС_!HOTDECK:_КОЛОДНАЯ) сохраняют в массиве переменную DONOR\_*ИМЯ*, показывающую, какие наблюдения выступили донорами. Вы указажете ее в п/к DONOR макроса !KO\_BORROW. См. ПРИМЕР 3 в описании !KO\_HDIMPUTE.

В !KO\_BORROW, если наблюдение-реципиент само является донором для каких-то других наблюдений, оно может передать им только те свои значения, которые были у него исходно. Таким образом, в массиве реципиентами заимствуются только первоначальные значения их доноров.

Макрос создает временные переменные с именами, оканчивающимися на *#$#$.* Рекомендуется поэтому, чтобы во входящем массиве данных не было переменных с именами, имеющими такое окончание.

ПРИМЕР 1.

!KO\_borrow vars= v1 v2 /donor= doncase /cap= imp\_ .



* Переменные, внутри которых наблюдения должны заимствовать значения у других наблюдений – *V1* и *V2*. Переменная DONCASE содержит номера наблюдений-доноров для текущих наблюдений как реципиентов. Заказано выдать новые переменные с приставкой *IMP\_*.
* По умолчанию обеих подкоманд IFRECVAL и IFDONVAL условие замещения такое: реципиентное значение замещается донорским, если первое есть пропуск, а второе валидно.
* Результат: пропуски замещены донорскими значеними согласно этому условию.

***Подкоманды***

**VARS**

Укажите поименно переменные, внутри которых делать заимствование значений между наблюдениями. Переменные могут быть либо числовые, либо текстовые, – все одного типа.

**DONOR**

Укажите имя числовой переменной, значения которой – номера наблюдений-доноров (для текущих наблюдений-реципиентов). Любые значения в этой переменной, кроме валидных целых положительных, макрос игнорирует (и, таким образом, текущее наблюдение не станет реципиентом).

**IFRECVAL, IFDONVAL**

Две подкоманды, ставящие условие, при котором заимствование значения у донора осуществится. IFRECVAL это условие «если значение наблюдения-реципиента, т.е. текущего наблюдения, в данной переменной VARS есть…». IFDONVAL это условие «если значение наблюдения-донора (т.е. упомянутого в переменной DONOR) в данной переменной VARS есть…». Одно из следующих ключевых слов:

ANY любое значение

VALID валидное значение

MISSING пропуск (system- или user-missing)

SYSMIS системный пропуск (употребимо при числовых VARS)

BLANK пустой (отсутствующий) текст, валидный или пропуск (употребимо при текстовых VARS)

Например, /IFRECVAL=ANY /IFDONVAL=ANY разрешает замену любого значения реципиента на любое значение его донора. По умолчанию/незаданию IFRECVAL считается MISSING; по умолчанию/незаданию IFDONVAL считается VALID; таким образом, если опустить обе подкоманды, произойдет импутация пропусков валидными данными.

Внимание, в SPSS Statistics ниже версии 21 не используйте условия VALID и MISSING для текстовых переменных, которые шире 8 байтов.

**CAP**

По умолчанию макрос изменяет сами переменные VARS. Если вы не хотите их трогать, а желаете, чтобы макрос создал новые переменные из них, укажите здесь приставку для имен новых переменных. Приставку можно взять в кавычки или апострофы (они рекомендуются, если приставку оканчиваете на разделитель-точку: ‘x.’). Если во входящем массиве уже есть переменные с такими именами, они будут переписаны.

***Особые режимы***

Макрос не слушается расщепленности массива (SPLIT FILE) и временных (под TEMPORARY) преобразований. Макрос игнорирует фильтрацию наблюдений (команды FILTER или USE) и взвешивание. Если надо запретить некоторым наблюдениям быть реципиентами или донорами – внесите соответствующую правку прямо в переменную DONOR. Всякая команда перед макросом, удаляющая часть наблюдений (например SELECT IF) радикально сказывается на результатах, т.к. сдвигает порядковые номера оставшихся наблюдений.