***Clustering tendency***

SPSS macros by Kirill Orlov

kior@akado.ru, ttnphns@gmail.com

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

All rights reserved

*Тенденция кластерности*. Некоторые методы разведки, которые позволяют предварительно судить до кластерного анализа, есть ли в данных кластеры. Блок-диагонализация матрицы расстояний между объектами может подсказать, сколько имеется кластеров. Статистика Хопкинса основана на симуляциях случайных бескластерных данных и сравнении их с наблюдаемыми данными.

*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.

* [Блок-диагонализация](#_МАКРОС_!KO_BLOCKDIAG:_БЛОК-ДИАГОНАЛ) нуждается в матрице расстояний. Этот метод имеет неявное родство с иерархической кластеризацией методом единичной связи.
* [Статистика Хопкинса](#_МАКРОС_!KO_HOPKINS:_СТАТИСТИКА) нуждается в данных «наблюдения - количественные признаки». И предполагает евклидово пространство и евклидовы расстояния. Статистика Хопкинса имеет целый ряд условий, зато она не связана имплицитно с каким-то методом кластеризации.

# МАКРОС !KO\_BLOCKDIAG: БЛОК-ДИАГОНАЛИЗАЦИЯ МАТРИЦЫ РАССТОЯНИЙ

Version 1, Mar 2023. Tested on SPSS Statistics 22, 27, 30.

!KO\_blockdiag matrix= *VAR1 to VAR80* /\*Столбцы, образующие тело матрицы расстояний (можно ч-з "to")

/id= /\*Опционально: числовая переменная-идентификатор наблюдений (рядов)

/method= VAT /\*Метод: VAT (тж п/у) или MDS

/poster= YES /\*"Постеризация" после перестановки: YES или NO (тж п/у)

/plot= /\*Тепловаяю карта: GREY (тж п/у) или RGREY;

/\*после можно добавить слово LABEL; либо NONE

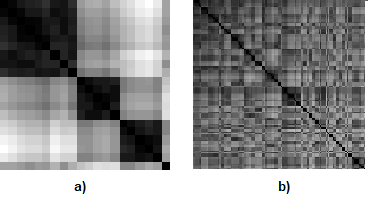
/bounds= /\*Пределы для тепловой шкалы на карте: AUTO (тж п/у),

/\*OBS или min max

/mds= /\*При METHOD=MDS: трансформация в MDS: SPLINE (тж п/у), ORDINAL, INTERVAL.

Минимум надо задать MATRIX.

Пусть есть матрица расстояний (различий) между объектами. Блок-диагонализация квадратной симметричной матрицы расстояний – это переупорядочивание ее рядов/столбцов – т.е. объектов – так, чтобы объекты с малыми расстояниями между ними разместились в матрице рядом друг с другом (т.е. их индексы стали близки). На тепловой карте такая переставленная матрица выглядит диагонально-блочной. Она тем более отчетливо, контрастно будет диагонально-блочна, чем сильнее присутствует тенденция к кластерности среди объектов, т.е. чем сильнее расстояния распадаются сами собой на «внутрикластерные» (малые) и «междукластерные» (большие). По тепловой карте, таким образом, можно приблизительно судить о наличии или отсутствии кластеров в данных, а также о числе кластеров, не делая самого кластерного анализа. Каждый кластер на тепловой карте после блок-диагонализации матрицы выглядит как блок на диагонали. Если кластеров нет, блок-диагонализация не покажет на тепловой карте ясных блоков.



**Рис. 1**. Тепловая карта блок-диагонализованной матрицы расстояний: a) с четкими кластерами в данных, b) без кластеров в данных.

Макрос делает блок-диагонализацию (переставляет ряды/столбцы) входящей матрицы расстояний, сохраняет полученную матрицу как новый безымянный массив данных и рисует тепловую карту.

**Алгоритм**

При METHOD=VAT макрос делает переупорядочивание рядов/столбцов матрицы расстояний алгоритмом VAT. VAT algorithm (“Visual Assessment of [Cluster] Tendency”) подробно описан в [1], также в [2,3]. Он тесно связан с алгоритмом построения минимального остовного дерева Прима взвешенного графа, и через это – имплицитно родствен иерархической кластеризации методом единичной связи, или ближайшего соседа [2]. Общей чертой этих алгоритмов является пошаговое наращивание остовного дерева/кластера/блока присоединением ближайших элементов.

Метод iVAT (“improved VAT”) [3] является надстройкой над VAT и по результату эквивалентен применению алгоритма Флойда–Уоршалла в варианте «определение легчайших проходов» к переупорядоченной матрице, выданной VAT. iVAT определенным образом заменяет некоторые расстояния в матрице другими ее расстояниями, уменьшая разнообразие расстояний в матрице. Идея iVAT проста: если две точки парно далеко друг от друга, но они опосредуются точечной цепью, все звенья (расстояния) в которой малы, то следует признать, что эти две точки «на самом деле» близки. Эффект iVAT-замены тот, что на тепловой карте (1) контраст между междукластерными и внутрикластерными расстояниями усилится, помогая визуально обнаружить кластеры-блоки; (2) повысится обнаруживаемость кластеров цепочечной структуры (включая сильно вытянутые, древовидные, кольцевидные). iVAT метод исполняется данным макросом, когда METHOD=VAT /POSTER=YES.

При METHOD=MDS макрос делает переупорядочивание рядов/столбцов матрицы расстояний, применяя многомерное шкалирование (SPSS-команда PROXSCAL) с взвешиванием расстояний. Это идея автора макроса (что не означает, что что она – новшество). Каждое расстояние *dij*, элемент матрицы, получает вес (важность) , где *Ri*– ранг значения *dij* в ряду *i*, *Rj* – ранг значения *dij* в ряду *j*, *R* – ранг значения *dij* в треугольнике матрицы. PROXSCAL исполняет картирование (ординацию) в пространстве размерности 1. Ряды/столбцы матрицы расстояний упорядочиваются вслед возрастанию полученных координат по этому измерению. Идея этого метода в том, чтобы принудить расстояния распределиться вдоль диагонали, отдавая в этом приоритет малым расстояниям. В методе VAT величина расстояния определяет очередность его подключения к остовному дереву. В методе MDS величина расстояния определяет его важность во влиянии на ординацию. В обоих методах конечным результатом становится то, что сгустки малых дистанций образуют блоки, нанизанные на диагональ матрицы.

METHOD=MDS /POSTER=YES применяет алгоритм Флойда–Уоршалла в варианте «определение легчайших проходов» к переупорядоченной матрице, выданной MDS-методом. Это производит такой же эффект, как iVAT после VAT.

Sources

1. Bezdek, J.C., Hathaway, R.J. VAT: a tool for visual assessment of (cluster) tendency // Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 – 2002 – Volume 3 – p. 2225–2230. [DOI:10.1109/IJCNN.2002.1007487]
2. Havens, T.C., Bezdek, J.C., Keller, J.M, Popescu, M., Huband, J.M. Is VAT really single linkage in disguise? // Annals of Mathematics and Artificial Intelligence. – 2009 – Vol. 55, article 237 [DOI 10.1007/s10472-009-9157-2]
3. Havens, T.C., Bezdek, J.C. An efficient formulation of the Improved Visual Assessment of Cluster Tendency (iVAT) Algorithm // IEEE Transactions on Knowledge and Data Engineering – 2012 – vol. 24, no. 5, p. 813-822 [DOI: 10.1109/TKDE.2011.33]

**Ограничения**

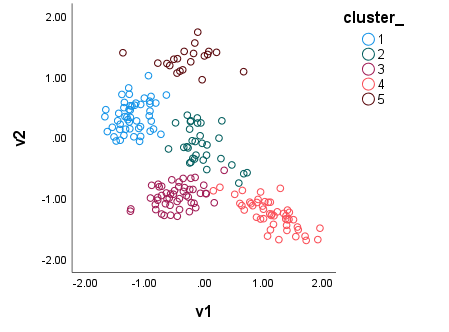
При METHOD=MDS составленная для анализа матрица – до 700 рядов/столбцов. При METHOD=VAT ограничений макрос не накладывает, но рекомендуемый размер матрицы – не более 1000, иначе тепловая карта будет рисоваться слишком долго. Вы всегда можете сделать анализ на случайной подвыборке наблюдений, если данные большие.

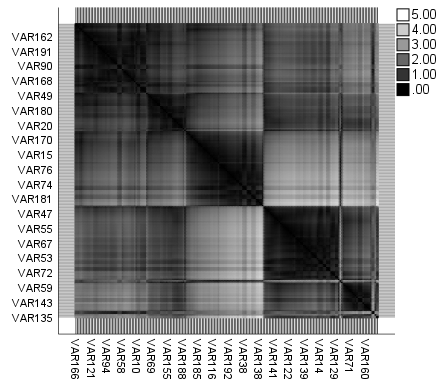
ПРИМЕР 1.

proximities v1 v2 /view= case /measure= seuclid /matrix= out(\*) /print= none.

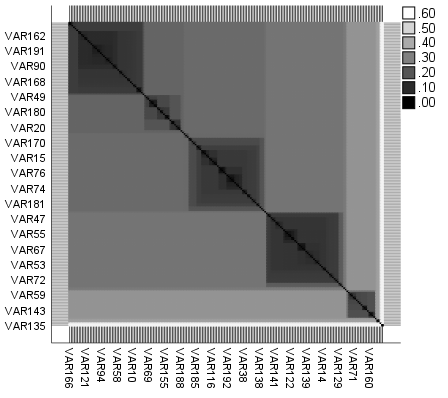
dataset name dist.

!KO\_blockdiag matrix= VAR1 to VAR199.





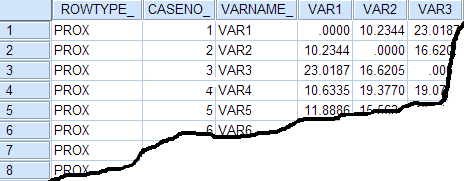
!KO\_blockdiag matrix= VAR1 to VAR199 /poster= YES.



* PROXIMITIES вычисляет из данных квадратные евклидовы расстояния между наблюдениями и сохраняет матрицу в новый массив, который называется *DIST*.
* Макрос исполняет блок-диагонализацию методом VAT и строит теплокарту. На ней можно распознать пять кластеров.
* Во втором пуске добавляется опция «постеризации». Пять кластеров проявились более отчетливо.

***Строение матрицы***

Массив данных должен быть матрицей попарных расстояний (различий, не сходств). Имена переменных – столбцов матрицы – до 8 байтов. Обязательно присутствие переменной VARNAME\_, именующей ряды в соответствие столбцам. Имена, являющиеся значениями этой переменной, должны быть написаны в том же регистре, как тождественные им имена среди имен переменных. Макрос не требует, чтобы ряды и столбцы шли в одинаковом порядке или чтобы их число и состав были полностью одинаковыми: макрос сам выберет из входящей матрицы одинаковые своими именами ряды и столбцы и соупорядочит их, чтобы таким образом составленная для анализа матрица имела квадратное диагонализованное строение. Переменная ROWTYPE\_ и прочие вспомогательные – не обязательны во входящей матрице.



***Подкоманды***

**MATRIX**

Укажите переменные рабочего массива, являющиеся столбцами собственно матрицы расстояний (различий). Вы можете указать все или только нужные столбцы и в произвольном порядке. Можно использовать “to” для задания диапазоном. Если у вас сходства, а не различия, то преобразуйте предварительно их в различия так, как сочтете нужным.

*Задание открытым диапазоном с помощью “?”*. В некоторых случаях возникает нужда указать диапазон, заключенный между парой переменных, которые сами не входят в диапазон. Используйте для этого “?” по краям. Например, *?VARNAME\_ to ENDVAR?* означает все переменные, находящиеся в массиве между переменными *VARNAME\_* и *ENDVAR*, не включая их самих. Для обозначения диапазона, открытого с одного конца, используйте “?” только у этого конца. Например: *?VARNAME\_ to VAR100* или *VAR1 to ENDVAR?*.

Т.к. ваши данные – различия, то «диагональные» значения - т.е. данные в ячейках на пересечении одноименных столбцов и рядов – должны быть нулями, а прочие («внедиагональные») значения должны быть неотрицательны; большее число отвечает большему различию.

ПРИМЕР 2.

temporary.

sample 0.2.

!KO\_blockdiag matrix= VAR1 to VAR100 /method= MDS /poster= YES .

* Команда SAMPLE временно (под TEMPORARY) отбирает случайно 20% рядов матрицы расстояний.
* Макрос берет столбцы *VAR1* до *VAR100*. Матрица, составленная макросом для анализа, состоит из рядов/столбцов, являющихся пересечением двух списков - отобранных рядов и отобранных столбцов.
* Макрос исполняет блок-диагонализацию методом MDS и «постеризацию» перед построением теплокарты.

**ID**

Опциональная числовая переменная-идентификатор наблюдений (объектов). Имя переменной до 8 байтов длиной. В переменной не может быть пропущенных значений.

**METHOD**

Укажите метод блок-диагонализации:

VAT - (тж. по умолчанию/незаданию) метод VAT.

MDS - многомерное шкалирование со взвешиванием расстояний исполняется SPSS-процедурой PROXSCAL. Она доступна в SPSS Statistics Professional Edition или в модуле Categories.

Оба метода часто дают очень похожие результаты.

**POSTER**

По умолчанию/незаданию и при POSTER=NO макрос делает только блок-диагонализацию, т.е. перестановку рядов/столбцов. При POSTER=YES он делает после этого «постеризацию». Тепловая карта при «постеризации» выглядит более контрастно, и обычно на ней легче увидеть кластеры (блоки). Кроме того, несколько повышается обнаруживаемость кластеров цепочечной структуры (включая сильно вытянутые, древовидные, кольцевидные). Иногда, впрочем, «постеризация» скрывает кластеры, которые близки друг к другу.

«Постеризация» при методе VAT известна как iVAT (“improved VAT”). «Постеризация» касается только теплокарты, она не влияет на сохраняемую матрицу. «Постеризация» не есть постеризация полученного изображения теплокарты, она есть контрастирование расстояний в матрице, по которой непосредственно строится теплокарта.

**PLOT**

Тепловая карта (heatmap) рисуется серо-полутоновой. При PLOT=GREY (тж. по умолчанию/незаданию) чем выше значение элемента, тем он ярче, а при PLOT=RGEY – наоборот, тем он темнее. После кл. слова вы можете добавить второе кл. слово LABEL, для оярлычения клеток значениями элементов. PLOT=NONE не рисует тепловую карту.

**BOUNDS**

Эта подкоманда не действует при PLOT=NONE. Задает границы для яркостной шкалы.

AUTO - (тж. по умолчанию/незаданию) позволить SPSS автоматически определить подходящие границы.

OBS - границы точно совпадают с наблюдаемыми минимальным и максимальным значениями в матрице.

*min* *max* - укажите границы вручную в виде двух чисел – минимума и максимума. Указывайте значения, в общем и целом сопоставимые со значениями матрицы.

Задание границ вручную означает, что яркостное отображение на рисунке вами фиксировано относительно величины содержащихся в матрице значений. Становится возможным сравнивать разные матрицы между собой по тону непосредственно.

**MDS**

Эта подкоманда действует при METHOD=MDS. Вы можете выбрать, метрическое или неметрическое многомерное шкалирование PROXSCAL использовать. Укажите INTERVAL (метрическое), ORDINAL (неметрическое) или SPLINE (промежуточное между метрическим и неметрическим). По умолчанию/незаданию, MDS=SPLINE.

***Особые режимы***

Макрос не слушается взвешивания (впрочем, он не берет в процедуру наблюдения с пропущенными и неположительными весами). Он не рассчитан на расщепленное состояние массива данных (SPLIT FILE). Макрос слушается команд, выбирающих наблюдения (SELECT IF, FILTER, USE), в том числе стоящих под командой TEMPORARY.

# МАКРОС !KO\_HOPKINS: СТАТИСТИКА ХОПКИНСА

Version 1, Mar 2025. Tested on SPSS Statistics 22, 27, 30.

!KO\_hopkins vars= *height weight blpress* /\*Анализируемые переменные; можно ч-з to

/jitter= /\*Опционально: сделать разгон данных: interval ALL или

/\*interval поименно переменные

/trim= /\*Опционально: % для отсечения выбросов, экстремумов

/z= /\*Делать анализ на стандартизованных переменных: NO (тж п/у) или YES

/numpc= MAX /\*Число гл компонент (до 10) или доля или MAX или NOPCA

/unifball= /\*Поджать периферию в данных: NO (тж п/у) или YES

/m= 0.1 /\*Число порождаемых точек или их доля от N

/power= /\*Параметр power: 0, 1, 2 или 3 (п/у =2)

/mult= /\*Опционально: параметр mult, число >=1

/repeat= 30 /\*Число раз порождать m точек и вычислять H (п/у =1)

/save= /\*Сохранение m точек и объектов (и данных): файл или заявленный массив

/id= /\*Опционально: ради SAVE идентификатор наблюдений.

Минимум надо задать VARS, NUMPC, M.

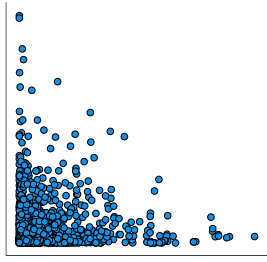
Макрос вычисляет статистику Хопкинса, являющуюся одной из мер тенденции кластерности в данных. Она основана на сопоставлении наблюдаемых данных и данных, симулированных случайно из равномерного распределения. Статистика Хопкинса позволяет судить, группируются ли наблюдения массива наблюдаемых данных в кластеры, или нет; следовательно, помогает решить, стоит ли делать кластерный анализ наблюдений; но она не говорит о том, сколько кластеров может быть в данных. Входящими данными являются количественные переменные. У вас должно быть не менее 100 наблюдений, но не слишком много, желательно не больше 2000.

Статистика Хопкинса (H) варьирует в пределах (0, 1). H на уровне 0.5 говорит о том, что кластеров нет (или, вернее, что они на уровне случайных группирований, возможных в случайной выборке из равномерного распределения). Н выше 0.7 позволяет подозревать реальные кластеры, а выше 0.8 решительно рекомендует делать кластерный анализ: явные кластеры присутствуют.

Данный макрос – одно из множества возможных осуществлений статистики Хопкинса, которые будут различаться тем, как именно и где создаются точки из равномерного распределения.

Статистика Хопкинса несколько прихотлива. Она либо данный макрос делают следующие допущения:

* Число объектов (наблюдений) больше 100, но не слишком много; для данного конкретного макроса рекомендуется примерно до 2000. Если у вас намного больше наблюдений, сделайте случайный отбор перед пуском макроса.
* В этом макросе число измерений в данных – до 10, и измерения не коррелируют. Используйте опцию главных компонент, встроенную в макрос, для редукции размерности и декорреляции ваших данных на входе. Редукция размерности имеет целью не только ускорение вычислений, но и использование только важных измерений, по которым потенциально различаются кластеры.
* Данный макрос полагает, что пространство данных – евклидово; расстояния в нем - евклидовы.
* Континуальные мерные данные. Используйте опцию разгона, встроенную в макрос, в случае дискретных/ликертовых данных.
* Макрос не годится для очень скошенных распределений, например таких как на **рис. 1**. Постарайтесь преобразовать данные к более симметричному или менее скошенному виду, а также используйте отсечение экстремальных хвостовых данных подкомандой TRIM.
* Кластеры по форме полагаются в данном макросе круглые или продолговатые – простые; не искривленные, звездообразные или кольцевидные.
* Для данного макроса «бескластерные» нуль-гипотезные данные – это случайные равномерные данные. Конкретно, случайное равномерное круглое облако, *p*-шар (в случае размерности *p*>1). Используйте UNIFBALL=YES для ваших колоколообразных данных вроде нормальных. Совет такой: попробуйте UNIFBALL=NO с вашими данными. Если H высоко, т.е. свидетельствует о кластерах, тогда попробуйте UNIFBALL=YES. Заключите, что есть кластеры, только если оба H высоки. Если же при UNIFBALL=NO H недостаточно высоко (не свидетельствует о кластерах), то пускать макрос в режиме UNIFBALL=YES излишне.



**Рис. 1**. Сильно скошенные данные не годятся для данного макроса.

ПРИМЕР 1.

temporary.

sample 1000 from 5420.

!KO\_hopkins vars= v1 to v8 /jitter= 1 ALL /numpc= .2 /m= .08.

!KO\_hopkins vars= v1 to v8 /jitter= 1 ALL /numpc= 3 /m= .08 /repeat= 10.

!KO\_hopkins vars= v1 to v8 /jitter= 1 ALL /numpc= 3 /m= .08 /repeat= 10 /unifball= YES.

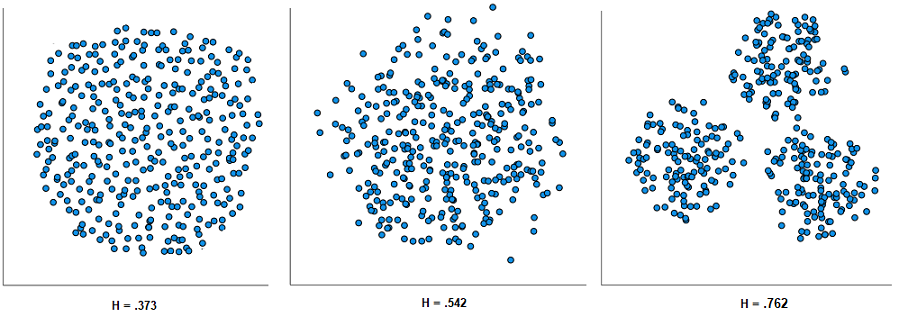
* В массиве 5420 наблюдений. Это больше, чем рекомендуется для макроса. Поэтому исследователь временно отбирает 1000 наблюдений случайно.
* Данные во всех *V1-V8* – это баллы по ликертовой шкале 1 2 3 4 5. Это дискретные, не континуальные значения, поэтому исследователь заказал разгон: JITTER= 1 ALL. M=.08 означает, что порождаться будет 80 случайных точек (8% от 1000).
* В первом пуске выделилось 4 главных компонент, потому что 4-я объясняет дисперсии всё еще больше, чем 1/5 от дисперсии 1-й компоненты, а 5-я – объясняет уже меньше. Получив результаты в Ouput, исследователь заметил, что первые три компоненты сравнимы между собой по величине, а 4-я уже гораздо слабее. Поэтому он решил переиграть, и во втором пуске заказал 3 главные компоненты (если в данных есть кластеры, они могут быть ответственны за образование этих сильных компонент[[1]](#footnote-1)). Он получил 10 значений статистики Хопкинса, со средней 0.801. Это говорит о наличии кластеров (полимодальности), но может и быть следствием хвостов унимодального распределения.
* Чтобы проверить последнее, исследователь пустил макрос с UNIFBALL=YES. Эта опция поджимает хвосты распределения данных, лишая облако данных периферийного лимба разреженности. Получившееся среднее значение статистики Хопкинса – 0.774, что тоже высоко. Исследователь сделал вывод, что в данных есть кластеры, достойные кластерного анализа.

**Алгоритм**

Статистика Хопкинса

Пусть нас интересует, есть ли кластеры в наборе *n* *объектов* (наблюдений массива данных) в *p*-мерном пространстве континуальных переменных. Мы порождаем из равномерного распределения *m* случайных *точек*. Область порождения должна совпадать с областью данных (т.е. мы «насыпаем» точки на облако данных). Вычисляем расстояние *ui* от каждой точки *i* до ее ближайшего соседа-объекта. Случайно отбираем *m* объектов из *n*. Вычисляем расстояние *wi* от каждого отобранного объекта *i* до его ближайшего соседа-объекта (из *n*-1 объектов). Статистика Хопкинса есть:

Когда в данных (т.е. среди объектов) нет кластеров, и будут близки по величине, потому что случайные точки и объекты будут разбросаны в пространстве сходно, случайно-равномерно. Поэтому H будет около 0.5. Если в данных есть кластеры, будет мало по сравнению с , потому что соседство между объектами будет в целом теснее, чем соседство между точками и объектами. H будет стремиться к 1. Возможно также, что H стремится к 0 – это имеет место, когда данные регулярны, являют собой равные промежутки между индивидуальными объектами. Но нам интересен прежде всего случай кластерности. Н выше 0.7 позволяет подозревать кластеры, а выше 0.8 решительно рекомендует делать кластерный анализ: явные кластеры присутствуют (**рис. 2**).

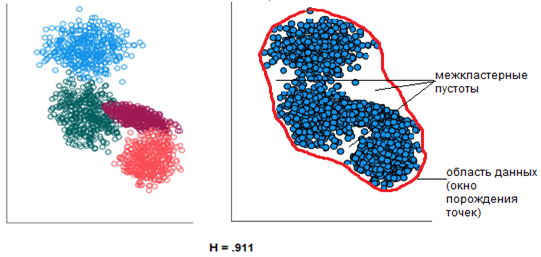


**Рис. 2.** В центре: случайные бескластерные данные; слева: более регулярные данные, чем случайные; справа: кластерные данные.

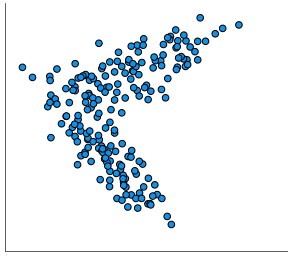
Под нуль-гипотезой, что объекты распределены случайно-равномерно, H следует бета-распределению с двумя одинаковыми *shape*-параметрами. Если мы знаем *shape*-параметр, мы можем выяснить статистическую значимость вычисленной H.

*m* должно быть << *n* (но не меньше 10). Рекомендуется *m* ≈ 0.1*n* или ≈ 0.05*n*. Дело в том, что бета-распределение имеет право, если у каждой точки свой ближайший сосед-объект, поэтому точек должно быть гораздо меньше объектов.

Основная теоретическая и практическая трудность – решить, что такое «облать данных» и где она кончается. Другими словами, проблема в окне порождения (sampling window) точек. Если мы «просыпем» точки за пределы области данных, мы увеличим и, следовательно, H, хотя кластеров может не быть. Важен не только размер, но и форма окна порождения. !KO\_HOPKINS пытается очертить окно порождения по форме и размеру облака данных (**рис. 3**) после придания измерениям равной дисперсии.



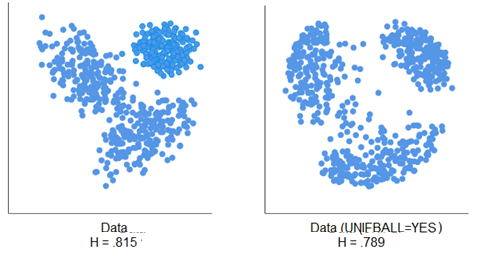
**Рис. 3**. Признаком кластерности для статистики Хопкинса является наличие разуплотнений между частями облака данных. Статистика Хопкинса ничего не говорит о числе кластеров, но только об их наличии. Статистика Хопкинса (как она реализована в !KO\_HOPKINS) годится для кластеров круглых или продолговатых, а не кластеров сложной, искривленной формы или кольцевых (**рис. 3a**).



**Рис. 3a**. С точки зрения статистики Хопкинса в реализации данного макроса, здесь скорее есть кластеры (они продолговатые и контактируют в одном месте), нежели их нет (одно кривое облако).

Операции макроса

1. Входящие количественные данные, *n* наблюдений (объектов), *p* переменных. *n* должно быть не менее 100 и желательно не больше 2 тыс.
2. Предобработка.
   1. Разгон данных. Заказывается, если значения в переменных дискретные (например – ликертовая рейтинговая шкала). Цель – превратить дискретные данные в континуальные.
   2. Переменные центруются (локус 0 помещается в центроид облака).
   3. Отсечение выбросов. Опционально. Заказанный процент объектов с наибольшими махаланобисовыми расстояниями до центроида изымается из данных. Теперь *n* – это оставшиеся объекты.
   4. z-стандартизация переменных. Опционально. Нужно, если переменные – в разных единицах измерения. z-стандартизация делается автоматом, если переменная единственная или если анализ главных компонент отменен.
   5. Анализ главных компонент (PCA). Рекомендуется. Делается, если *p*>1. Целью PCA является: (i) снижение размерности, (ii) декоррелирование, (iii) придание измерениям равной важности[[2]](#footnote-2). Переменные заменяются ортогональными главными компонентами, которые стандартизованы (их средние 0 и дисперсии 1). Отныне *p*, число измерений, – это число главных компонент (которых может быть выделено меньше, чем есть переменных), а значения данных – это есть компонентные баллы. Число измерений может быть максимум 10.
   6. Преобразование «нормальные данные в равномерные» (UNIFBALL=YES). Опционально. Эта операция поджимает периферию облака к центру. Если данные есть нормальное облако, то в результате поджимания выходит равномерное облако, причем при *p*>1 это равномерный *p*-шар. UNIFBALL=YES нужно для дополнительной проверки на кластеры (**рис. 4**).
3. Порождение и отбор случайных точек из равномерного распределения.
   1. Центр (локус 0) помещается на полпути между центроидом данных и серединой диапазона в данных.
   2. Порождение на области данных (объектов) и около нее случайных точек из коробчатого (прямоугольного) равномерного распределения.
   3. Оконтуриваине (projection-rejection методом) облака данных: отбор точек, чтобы их облако облекало облако объектов.
4. Вычисление статистики Хопкинса.
   1. Случайное взятие *m* точек (из отобранных точек) и *m* объектов (*m* обычно 0.1*n* или меньше).
   2. Вычисление евклидовых расстояний до ближайших соседей: от взятых точек до объектов и от взятых объектов до объектов.
   3. Вычисление H по формуле.



**Рис. 4**. Нативное облако с кластерами и искажение его под UNIFBALL=YES.

Нуль-облако

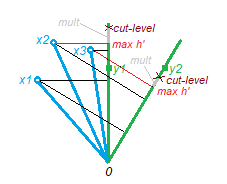
Нуль-облако, бескластерное облако, это популяция объектов без кластеров, но, с другой стороны, и без регулярной структуры объектов (вроде решетки или паутины с одиночными объектами в узлах). Статистика Хопкинса принимает за нуль-облако случайное облако из равномерного распределения. Макрос !KO\_HOPKINS (поскольку использует оконтуривание, диктуемое из центра, идущее по кругу), уточняет, что при размерности *p* 2 и более это есть равномерный *p*-шар (а не *p*-куб). H-статистика калибрована в макросе так, что она ожидается 0.5 на этом нуль-облаке.

Порождение случайных точек (2.2)

Точки порождаются из прямоугольного равномерного распределения RV.Uniform(min·*mult*, max·*mult*), где min и max – наблюдаемые минимум (отрицательное число) и максимум (положительное число) в данных, а припуск *mult≥*1 объяснен в дальнейшем.

Projection-rejection оконтуривание (2.3)

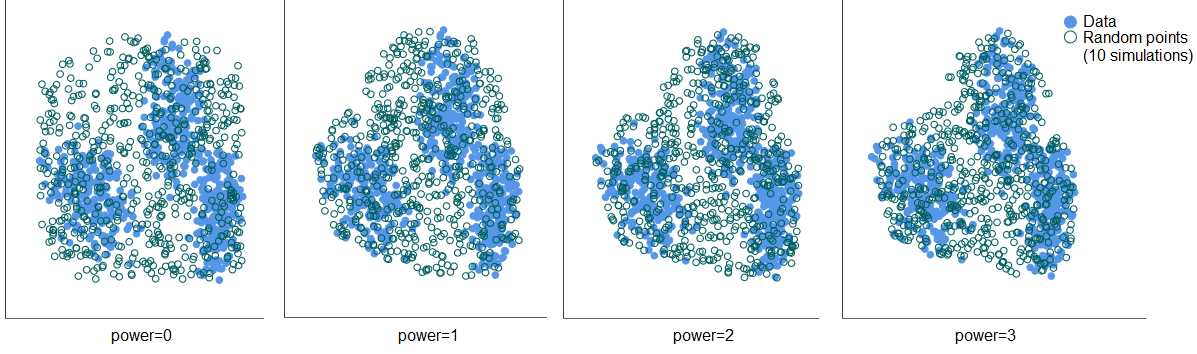
Основной вызов в предприятии по получению статистики Хопкинса – это как облечь облако наблюдаемых данных случайными точками. Интуитивно, точки должны занимать область данных и ограничиваться ей. Др. словами, (1) там, где есть объекты, точки и объекты должны лежать в среде друг друга: перемешанно, их фазы должны совпадать. (2) Также, точки должны лежать в межкластерном пространстве*.* Оконтуривание projection-rejection методом, принятым в данном макросе, пытается приблизительно решить эту задачу.



**Рис. 5**. Идея projection-rejection оконтуривания. Три объекта *x* и две порожденные точки *y*.

Объекты *x* проецируются на лучи точек *y* (**рис. 5**). Длина проекции , где это луч и отстояние *x* от центра 0, и это косинус угла между лучом , несущим объект, и лучом , несущим точку. cos<0 принимается за 0. Степень *power* объяснена ниже. Получив проекции всех объектов данных *x*1, *x*2, … на луч, несущий данную точку *y*, выберем максимальную проекцию из них, . И уровень отсечения есть эта проекция, умноженная на коэффициент *mult≥*1, объясненный в дальнейшем. Если , отклонение точки *y* от центра, меньше , то точка принимается как лежащая на области данных, иначе она отвергается как выходящая за область данных.

На **рис. 5** *power*=1, поэтому проекции отвесны. *power*=2 делает проекции косыми, наклоненными внутрь (к локусу 0), т.е. уменьшает их, и уменьшение будет тем сильнее, чем меньше косинус, т.е. чем больше угол между лучом объекта и лучом точки. *power*>1 ведет к более тесному, скупому оконтуриванию (**рис. 6**), поскольку выводит из игры в данном месте пространства объекты, углово далекие от этого места. *power*=0, напротив, превращает проекцию в поворот луча в луч , так что очень отдаленные объекты сказываются на судьбе точки. Мы рекомендуем использовать *power* 1 или 2. 2 – по умолчанию.



**Рис. 6**. Параметр *power* и его влияние на форму оконтуривания данных точками.

Параметр *mult*

С ростом *p/n* оконтуривание projection-rejection методом ведет, к сожалению, к размежеванию фаз объектов и точек. Причиной является проклятие размерности. В случайных данных высокой размерности объекты стремятся лежать на примерно равных дистанциях друг от друга, и каждый объект стремится быть внешним, принадлежать выпуклой оболочке облака данных, когда объектов немного. Получается, что внутри облака данных почти нет объектов[[3]](#footnote-3). Но там будут точки, отобранные оконтуриванием. Точки в условиях относительно высокого *p* и небольшого *n* будут кнутри от объектов. И точки будут в целом ближе к ближайшим объектам, чем объекты к ближайшим объектам. Это приведет к занижению H. (Вообще, projection-rejection оконтуривание имеет целью изъять точки из внекластерного пространства и оставить их в межкластерном пространстве. Но в высокой размерности, в связи с его разреженностью, разница между межкластерным и внекластерным регионами быстро сходит на нет.)

Чтобы компенсировать указанную тенденцию разделения фаз, можно ввести при оконтуривании *припуск* для точек: пусть точке разрешено лежать периферийнее (см. **рис. 5**), чем то место максимальной проекции, которое вычисляет projection-rejection оконтуривание. Этот припуск может быть простым константным для всех точек множителем *mult*, величина которого зависит от *p* и *n* (и от *power*). Множитель подбирается чисто эмпирически таким, чтобы в симуляциях со случайными данными из равномерного распределения H получилась со средней арифметической, близкой к 0.5 (**табл. 1**). Совпадения фаз не будет: точки будут занимать теперь наоборот, несколько больший объем пространства, чем объекты, но H будет центрована на величине 0.5.

**Табл. 1**. Подобранные значения *mult*, при которых среднее значение H, полученное в симуляциях случайных бескластерных данных (равномерный *p*-шар, при *p*>1), близко к 0.5.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *p*: | 1 | 2 | 3 | 4 | 5 | 7 | 10 |
| *n*: |  |  |  |  |  |  |  |  |
| 3000 |  | 1 |  |  |  |  |  |  |
| 2000 |  | 1 | 1.01/1.015 | 1.03/1.048 | 1.06/1.1 | 1.094/1.165 | 1.173/1.317 |  |
| 1000 |  | 1.003 | 1.02/1.02 | 1.043/1.07 | 1.082/1.133 | 1.122/1.217 | 1.212/1.39 | 1.345/1.645 |
| 700 |  | 1.005 | 1.024/1.03 | 1.055/1.084 | 1.098/1.16 | 1.144/1.249 | 1.238/1.44 | 1.379/1.695 |
| 500 |  | 1.008 | 1.03/1.04 | 1.068/1.105 | 1.114/1.188 | 1.166/1.285 | 1.267/1.485 | 1.413/1.745 |
| 300 |  | 1.013 | 1.04/1.06 | 1.089/1.14 | 1.146/1.24 | 1.203/1.348 | 1.32/1.564 | 1.466/1.822 |
| 200 |  | 1.019 | 1.06/1.08 | 1.116/1.175 | 1.177/1.288 | 1.24/1.41 | 1.367/1.634 | 1.52/1.886 |
| 100 |  | 1.038 | 1.1/1.13 | 1.18/1.258 | 1.25/1.399 | 1.326/1.537 | 1.463/1.77 | 1.624/2.004 |

Над дробью – значение для *power*=1; под дробью – значение для *power*=2.

Мы аппроксимировали значения в **табл. 1**, зависящие от *n* и *p*, регрессионно. Мы использовали программу DataFit верс. 9, позволяющую подгонять множество регрессионных моделей, и остановились на уравнении вида

Y = a+b\*ln(*n*)+c\**p*+d\*ln(*n*)^2+e\**p*^2+f\*ln(*n*)\**p*+g\*ln(*n*)^3+h\**p*^3+i\*ln(*n*)\**p*^2+j\*ln(*n*)^2\**p*

для случая *p*>1, и на уравнении вида

Y = a+b\*ln(*n*)+c\*ln(*n*)^2+d\*ln(*n*)^3+e\*ln(*n*)^4

для случая *p*=1.

Сами уравнения получились такими. Этими уравнениями макрос вычисляет параметр *mult*.

Для *p*>1, *power*=1:

*mult* = 1.8001412953476-.37821110173515\*ln(*n*)+.200639641499899\**p*

+5.34921458279123E-02\*ln(*n*)^2-2.29345019432612E-03\**p*^2

-3.37379264269758E-02\*ln(*n*)\**p*-2.28570712583439E-03\*ln(*n*)^3

-2.85869384260207E-04\**p*^3+1.21163113175788E-03\*ln(*n*)\**p*^2+7.63762113783788E-04\*ln(*n*)^2\**p*

(R^2 = 0.9997846286)

Для *p*>1, *power*=2:

*mult* = 1.55402545812428-.285142518531347\*ln(*n*)+.316937136127879\**p*

+3.54656315363041E-02\*ln(*n*)^2-4.85182536412298E-03\**p*^2

-4.64194001390029E-02\*ln(*n*)\**p*-8.44286652372263E-04\*ln(*n*)^3

-7.80112278128893E-04\**p*^3+3.03660728431481E-03\*ln(*n*)\**p*^2-2.99246137741637E-04\*ln(*n*)^2\**p*

(R^2 = 0.9997016488)

Для *p*=1:

*mult* = 2.48919971705474-.843966420677826\*ln(*n*)+.183812420447206\*ln(*n*)\*\*2-1.80707757504798E-02\*ln(*n*)\*\*3

+6.71576972339337E-04\*ln(*n*)\*\*4

(R^2 = 0.9998991173)

Для *power*=0 или 3 макрос не вычисляет *mult*. Пользователь должен сам задать *mult*.

Как было показано ранее, параметр *mult*, т.е. припуск, используется дважды. Во-первых, при порождении исходной «коробки» точек из равномерного распределения. Во-вторых, при отборе точек projection-rejection оконтуриванием.

Значимость H

Sig. нулевой гипотезы, что в данных нет кластеров, против односторонней альтернативной гипотезы, что в данных есть кластеры, = 1-CDF.Beta(H,*shape*,*shape*), где *shape* – параметр в бета-распределении. Теоретически, в условиях бесконечного пуассонова процесса в пространстве, *shape* = *m*. В реальных условиях нашего макроса, где имеется ограниченное в пространстве нуль-облако в форме *p*-шара, *shape* лишь приближено к *m* и зависит от *n* и *p*. В **табл. 2** показаны несколько замеренных нами (в SPSS-процедуре P-P plot) значений *shape* при *m* = 25.

**Табл. 2**. *Shape*-параметр при *m* = 25.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *p*: | 1 | 2 | 3 |
| *n (m/n)*: |  |  |  |  |
| 1000 (.02) |  | 25.0 | 23.2 | 21.7 |
| 500 (.04) |  | 24.4 | 22.9 | 21.2 |
| 200 (.10) |  | 22.8 | 22.0 | 21.0 |

Как можно видеть, с увеличением *m/n*, а также с увеличением *p shape* падает. Можно смоделировать зависимость *shape* от *m*, *n* и *p*, но в настоящее время эта работа не сделана. Поэтому макрос не вычисляет Sig.

Чтение о статистике Хопкинса

* Hopkins, B., Skellam, J.G. A new method for determining the type of distribution of plant individuals // Annals of Botany. – 1954 - 18(2). - pp. 213-227.
* Adolfsson, A., Ackerman, M., Brownstein, N.C. To cluster, or not to cluster: An analysis of clusterability methods // Pattern Recognition. – 2019. – 88. – p. 13–26.
* Panayircit, E., Dubes, R.C. A test for multidimensional clustering tendency // Pattern Recognition. – 1983. – 16. – p. 433-444.
* Banerjee, A., Dave, R.N. Validating clusters using the Hopkins statistic // FUZZ-IEEE 2004. - 25-29 July, 2004, Budapest, Hungary.
* Cross, G.R., Jain, A.K. Measurement of clustering tendency // Theory and Application of Digital Control. - New Delhi, India. – 1982.

**Быстродействие**

Скорость зависит прежде всего от размерности пространства *p* и от числа отбираемых точек *m*. В пространстве высокой *p* (например, 10) точки отбираются projection-rejection методом крайне медленно, и на накопление *m* точек, особенно когла *m* велико, уходит много времени. Также, скорость зависит от числа объектов *n*.

***Подкоманды***

**VARS**

Анализируемые переменные. Полный список и/или через “to”. Это должны быть количественные признаки. Если в данных есть пропущенные значения, макрос исключит их списочно, т.е. всё наблюдение будет исключено из анализа, если оно есть пропуск хотя бы в одной переменной.

**JITTER**

Предобработка 1. Разгон объектов. Используйте, если ваши данные дискретны, например это рейтинговые баллы по шкале типа Ликерта. Разгон данных превращает дискретные значения в непрерывные. Непрерывность переменных является необходимым условием статистики Хопкинса.

Укажите число – интервал между градациями шкалы, и после этого поименный список дискретных переменных из VARS, либо ALL (что значит «все переменные VARS»). Например, JITTER= 1 ALL означает, что все переменные VARS – дискретные, с интервалом в один балл между смежными градациями шкалы. JITTER= 1 VAR1 VAR2 VAR5 означает, что из VARS дискретными являются только переменные VAR1, VAR2, VAR5 (и во всех них интервал между смежными градациями шкалы равен 1), а остальные переменные VARS – непрерывные по своим значениям.

Важно:

1. интервал можно указать только один. Если у вас в разных переменных – шкалы с неодинаковыми по ширине интервалами (например, есть шкала 1 2 3 4 и есть шкала 10 20 30), преобразуйте переменные сначала, сделав интервал везде одинаковой ширины.
2. имена переменных, указываемых поименно в п/к JITTER, должны быть до 8 байтов длиной и написаны в том же регистре букв, в каком они написаны в массиве данных.

**TRIM**

Предобработка 2. Отсечение заданного процента выбросов/экстремумов. Выбросы имеют тенденцию завышать статистику Хопкинса. Укажите процент объектов, например TRIM=2.5. Тогда 2.5% объектов (наблюдений) с наибольшим махаланобисовым расстоянием от центроида данных будут исключены из анализа. Процент должен быть выше 0 и не выше 10.

**Z**

Предобработка 3. Z=YES стандартизует переменные перед анализом главных компонент. Стандартизация нужна, если единицы измерения у переменных разные. Если анализ главных компонент отменен (NUMPC=NOPCA), стандартизация переменных делается даже в том случае, если она не заказана (Z=NO, тж. по умолчанию).

**NUMPC**

Предобработка 4. Подкоманда обязательна, если VARS более одной переменной. Анализ главных компонент (PCA) замещает входящие переменные стандартизованными главными компонентами. Целью является снизить размерность (т.к. макрос разрешает максимум 10 измерений в данных) и декоррелировать измерения (что важно для процедуры оконтуривания данных – см. «Алгоритм»). Отныне «данные»– это будут главные компоненты.

Укажите одно из следующего:

*число компонент* - целое число (не больше 10), сколько выделить главных компонент.

*доля* - десятичное дробное число >0 и <1. Выделить все главные компоненты с собственными числами >*доля*·*λ1*, где *λ1* это собственное число 1-й компоненты.

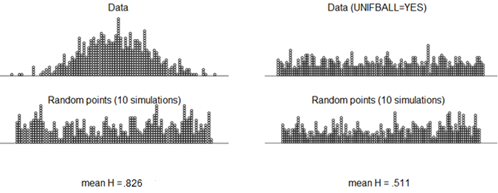
MAX - выделить все ненулевые главные компоненты (но не больше 10).

NOPCA - не делать анализ главных компонент. Эта опция предполагает, что переменных VARS – до 10, и они не коррелируют или почти не коррелируют.

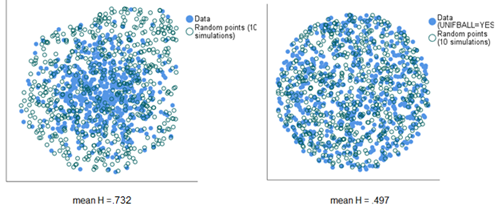
Сделав PCA, макрос показывает собственные числа и процент объясненной дисперсии, так что вы можете решить, сколько компонент выделить (оставить) при следующем пуске макроса. Надо выделять только сильные компоненты. Например, NUMPC=0.25 – неплохой выбор, он оставляет компоненты, стандартное отклонение у которых больше, чем половина стандартного отклонения у 1-й компоненты. Все выделенные компоненты макрос рассматривает как одинаково важные измерения – стандартизуя их баллы.

**UNIFBALL**

Предобработка 5. По умолчанию и при UNIFBALL=NO макрос анализирует неискаженное облако данных. При UNIFBALL=YES он анализирует облако данных с поджатой периферией (хвостами). Чем дальше отстоял объект от центроида, тем на большее расстояние он смещается к центроиду. Если облако данных было нормальным распределением без кластеров, UNIFBALL=YES превращает его в равномерное распределение (**рис. 7**); в случае *p*-мерного нормального распределения это будет равномерный *p*-шар (**рис. 8**). См. тж. **рис. 4**.



**Рис. 7**. Нормальное распределение у данных Data (слева) и оно же, превращенное в равномерное (справа).



**Рис. 8**. Нормальное 2-мерное распределение у данных Data (слева) и оно же, превращенное в 2-мерный равномерный шар (справа).

Статистика Хопкинса, как она реализована в макросе, базируется на нуль-гипотезе, что данные исходят из равномерного распределения, шаровидного при *p*>1. Для макроса данные из нормального и вообще колоколообразного унимодального распределения – это данные не бескластерные, а с одним кластером: статистика Хопкинса будет повышена. Поэтому, если в режиме UNIFBALL=NO вы видите высокую статистику Хопкинса, то не спешите заключить, что в данных есть *кластеры*. Пустите макрос второй раз, теперь в режиме UNIFBALL=YES. Если и в этом режиме статистика Хопкинса высокая, тогда сделайте вывод, что данные состоят из кластеров.

Статистика Хопкинса под UNIFBALL=YES часто ниже, чем под UNIFBALL=NO. Если под UNIFBALL=YES она превосходит 0.6 (притом что под UNIFBALL=NO она выше 0.7), вы можете подозревать наличие не ясно выраженных кластеров.

**M**

Число случайных точек, порождаемых/отбираемых в области данных. Укажите целое не меньше 10. Либо укажите десятичную дробь не больше 0.1. Эта дробь есть доля от *n*, числа объектов (наблюдений) в данных. Обычно долю задают от 0.05 до 0.1.

**POWER**

Степень, в какую возводить косинус при projection-rejection отборе точек (см. «Алгоритм»). Укажите число 0, 1, 2 или 3. По умолчанию, POWER=2. POWER 0 или 3 не рекомендуем. POWER игнорируется, если переменная VARS единственная или экстрагируется единственная компонента.

**MULT**

Необязательная подкоманда, которой вы можете задать свое значение параметра *mult* (см. «Алгоритм»). По умолчанию, *mult* вычисляется макросом по формуле, кроме когда POWER= 0 или 3.

**REPEAT**

Сколько раз порождать/отбирать точки и вычислять статистику Хопкинса. Укажите целое положительное число. По умолчанию, REPEAT=1.

**SAVE**

Это подкоманда для сохранения данных после их предобработки (какая была заказана), сохранения отобранных для вычисления статистики Хопкинса *m* точек и *m* объектов. Укажите .SAV-файл для сохранения либо имя объявленного массива.

В файле/массиве будет переменная *SOURCE\_* со значениями 0, положительными и отрицательными целыми. 0 помечает *n* наблюдений данных, т.е. все объекты. 1 помечает *m* сгенерированных и отобранных случайных точек для 1-й из REPEAT раз вычисленной статистики Хопкинса, а -1 помечает *m* случайно отобранных объектов для вычисления ее. 2 и -2, соответственно, помечают точки и объекты, на которых вычислена 2-я статистика Хопкинса. И так далее, до цифры REPEAT.

ПРИМЕР 2. Видеть облако данных и порожденные точки.

dataset declare savedata.

!KO\_hopkins vars= v1 to v8 /z= YES /numpc= 2 /m= .08 /repeat= 10 /save= savedata.

dataset name h.

dataset activate savedata.

select if source\_>=0.

recode source\_ (0=0) (else=1).

value labels source\_ 0 'data ' 1 'points '.

graph /scatterplot(bivar)= pc1 with pc2 by source\_.

* Декларируется массив *SAVEDATA*. Макрос сохраняет материалы, использованные для вычисления 10 штук статистики Хопкинса, в этот массив.
* Массив со статистикой Хопкинса получает имя *H*.
* Активируется массив *SAVEDATA*. SELECT IF удаляет из него отобранные для вычислений объекты, оставляя только сами данные (все объекты) и порожденные/отобранные точки. RECODE перекодирует все номера 1, 2, …, 10 (вычислялось 10 штук статистики) в 1.
* Строится диаграмма рассеяния.
* Данные – это данные после предобработки. В данном случае предобработка заключалась в стандартизации входящих переменных и выделении двух главных компонент, их баллы стандартизованы.

**ID**

Вы можете указать числовую переменную-идентификатор наблюдений в вашем входящем массиве. Идентификатор имеет смысл, когда вы задали SAVE. Если в данных были пропуски или если вы использовали п/к TRIM, то в сохраненных п/к SAVE данных будет меньше наблюдений (*SOURCE\_*=0), чем во входящем массиве. Идентификатор поможет вам отследить наблюдения.

***Особые режимы***

Макрос не слушается взвешивания (впрочем, он не берет в процедуру наблюдения с пропущенными и неположительными весами). Он не рассчитан на расщепленное состояние массива данных (SPLIT FILE). Макрос слушается команд, выбирающих наблюдения (SELECT IF, FILTER, USE), в том числе стоящих под командой TEMPORARY.

1. Причем можно даже предположить, что кластеров 3+1, т.е. 4. [↑](#footnote-ref-1)
2. Мы сознаём, что уравнивание дисперсий измерений есть некоторое искажение данных. Кластеры и межкластерные пустоты могут поменять свою форму. Но вклад измерений надо уравнять, потому что формула статистики Хопкинса отводит измерениям равный вклад; также, используемое макросом projection-rejection оконтуривание подразумевает примерно равные дисперсии у измерений. [↑](#footnote-ref-2)
3. В предельном случае, когда *p*=*n*-1, мы практически всегда имеем симплекс, фигуру без объектов «внутри». [↑](#footnote-ref-3)