***Generate random clusters, mixtures***

SPSS macros by Kirill Orlov

kior@akado.ru, ttnphns@gmail.com

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

All rights reserved.

*Random cluster/mixture data.* Creation of random data consisting of clear clusters or mixtures (fuzzy clusters). Can make these clouds round or elongated, gaussian or platykurtic, regulate sizes and bodily closeness among them. A separate macro randomly rotates data in space.

*Read “*[*About SPSS macros*](https://www.spsstools.net/en/KO-aboutmacros)*” what are they and how to run them.*

*The “Protected directory” error.* Some of the macros described in the current document write temporary files to hard disc. If you don't have full Administrator rights of your computer, it may cause error saying, among things: *“SPSS Statistics cannot access a file... specifies a protected directory...”*, meaning that the default directory the macro wants to use is protected on your PC. To solve the problem, in Syntax window issue command: CD 'myfolder'., where 'myfolder' is the path/name of some folder where you are allowed to save files to.

* [!KO\_GENCLU](#_МАКРОС_!GENCLU:_ПОРОЖДЕНИЕ_СЛУЧАЙНЫ) generates clusters and mixtures, spherical or spheroid (oblong in one direction).
* [!KO\_ROTCLU](#_МАКРОС_!ROTCLU:_СЛУЧАЙНЫЙ_ПОВОРОТ_() rotates randomly data in space or deprives them from rotatedness.

# MACRO !KO\_GENCLU: GENERATING RANDOM CLUSTER/MIXTURE DATA

Version 2, May 2014 (Version 1, Apr 2010). Tested on SPSS Statistics 17, 20, 27.

!KO\_genclu k= *3* /\*Number of clusters (from 2 to 100)

/p= *2* /\*Number of variables (from 1 to 50)

/n= *100* /\*(Approximate) number of points in cluster: one number, or two - min and max

/centers= EVEN /\*Initial cluster centres: RANDOM (create them random, default),

/\*EVEN (create them approximately evenly-spaced),

/\*or quoted external SAV-file name (centers are specified manually)

/spacing= *.8* /\*Closeness between clusters bodies: number from 0 (tightly adjacent clusters)

/\*to 1 (widely apart clusters), or RANDOM (define randomly, default)

/sizing= *0* /\*Sameness of clusters physical sizes: number from 0 (same sized)

/\*to 1 (strongly different-sized), or RANDOM (define randomly, default)

/flat= /\*Flatness of clusters: number from 0 (normal kurtosis, default) to 1

/\*(flat kurtosis, cluster is a ball), or RANDOM (define randomly for

/\*each cluster)

/elong= STRETCH *1 3* /\*Optionally, for P>1: make clusters elongated - stretched (STRETCH)

/\*or extended (EXTEND); after the keyword specify

/\*2 elongatedness parameters (numbers from 1 to 4)

/rotat= /\*For elongated clusters: their rotation arbitrary (FREE, default),

/\*approximately in the same direction (RESTR) or don’t rotate at all (NONE)

/fix= YES /\*Bring obs centroids of the generated clusters exactly to the initial centers:

/\*YES or NO (default)

/clear= LDA /\*Optional: make clusters clear (classify the points) by method:

/\*KNN (k nearest neighbors), LDA (discriminant analysis),

/\*KMC (K-means, classify only)

/weed= *1* \*For clear clusters: weeding of boundary zones - number from 0 (don’t do, default)

/\*to 1 (do fully).

Minimal specification K, P, N.

The macro generates random cluster/mixture data. You can request to generate from 2 to 100 clusters; number of variables (space dimensionality) from 1 to 50. The macro outputs a new unnamed working dataset with variable *CLUSTER\_* (cluster number) and variables *V1, V2,* and so on (point coordinates).

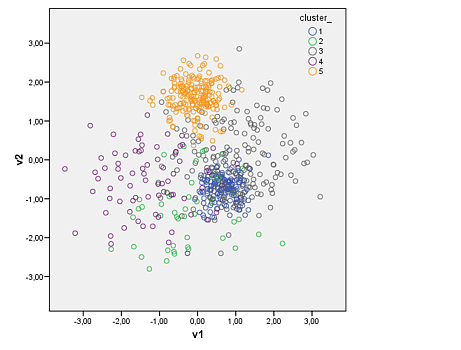
Initial cluster centres – around which the random points are generated – you can ask the macro to define them by itself or you can specify your own as you like. You can allow the macro to define closeness between clusters and sizes of clusters by itself or regulate these parameters according to your wish. By default, clusters are created round (spherical) with normal distribution of points, but you can demand clusters elongated to varying and random degree, as well as flattened (lessened kurtosis up to uniform distribution). Finally, you can create clusters fuzzy (i.e., mixtures) or clear (not allowed to intersect). For clear clusters, it is possible to order to do “weeding” of the zones of contiguity, to lessen the density of points there.

Elongated clusters are currently available spheroid, not ellipsoid. Distribution is clusters is not skewed.

To manage random number seed in SPSS Statistics, use menu Transform – Random Number Generators or corresponding syntax.

EXAMPLE 1.

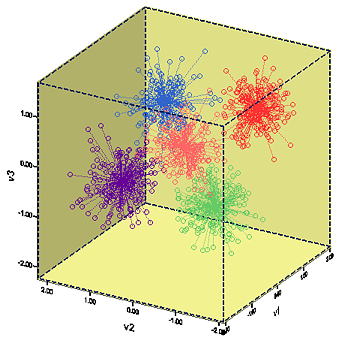
!KO\_genclu k= 5 /p= 2 /n= 50 200.



* In a 2-dimensional space, 5 round clusters are generated, with number of points from 50 to 200 in each. Distances between cluster centres, physical size of clusters (variance magnitude in them) are allowed to be selected randomly by the macro. The clusters are allowed to remain overlapping (fuzzy).

EXAMPLE 2.

!KO\_genclu k= 5 /p= 3 /n= 200 /centers= EVEN /spacing= .8 /sizing= 0 /clear= KMC /weed= 1.



* In a 3-dimensional space, 5 round clusters are generated, 200 points per each. Clusters are relatively evenly scattered over the space (CENTERS=EVEN), they are the same physical size (SIZING=0) and stand quite away from each other (SPACING=0.8). The clusters are requested to make nonoverlapping, clear (CLEAR=KMC) and to weed boundary zones of density (WEED=1). However, the last two subcommands hardly affected any much, because the clusters, standing in this case quite far away form each other, practically did not intersect when generated.

***Subcommands***

**K**

Number of clusters *k*. Specify integer from 2 to 100.

**P**

Number of variables (space dimensionality *p*). Specify integer from 1 to 50.

**N**

Number of points (cases) in a cluster. Specify one or two positive integers. If one integer, all the clusters will be equal by the number points. If two integers, the clusters will vary within these two limits by the number of their points (it doesn’t matter which number, the lesser or the greater, you indicate first).

Be aware that if you are ordering to weed the boundary zones (see WEED s/c), the number of points in clusters may in the end turn less than that you’ve specified, because points are partly deleted at weeding (and are deleted heavier the stronger clusters overlap by their margins, - see SPACING s/c).

**CENTERS**

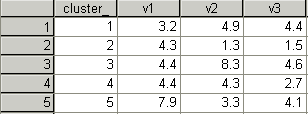
Specification of cluster initial centres, i.e. centres around which random data points will be generated. Choose one of the three:

RANDOM - (also default/unspecifying the subcommand) initial centres are created as random points in *p*-dimensional space. One neighboring centres may occur close to each other, while other neighboring – at a distance to each other. In other words, the clusters, as the result, can themselves appear elements of clusters of a higher order.

EVEN - initial centres are created random, but approximately evenly scattered over the space. Use this variant if you want clusters to be not mergeable with each other in clusters of a higher order.

*Имя файла* - initial centres are specified by the user. Indicate (in quotes or apostrophes) path/name of an external SAV file containing the initial centres’ coordinates.

The external SAV file with the centres’ coordinates must have the build as shown on the picture: should have numeric variable *CLUSTER\_* with cluster ordinal numbers 1, 2, …, *k* (where *k* is the same number you indicate in K s/c). And centre coordinates in variables V1, V2, …, V*p* (where *p* is the same number you indicate in P s/c). No other variables of extra rows should be in the file. Violating these rules will cause an error.



**SPACING**

Parameter defining tightness of fit of neighbouring clusters to each other. (If clusters are allowed to intersect – e.g. CLEAR s/c is not specified – then this is the degree neighbouring clusters overlap one another with their margins, forming a mixture.) Specify one of the following:

*Number from 0 to 1* - 0 creates clusters, all neighbouring (i.e., nearest) of which will be co-adjacent tightly. On the other hand, 1 creates clusters, all neighbouring of which will stand far away from each other. You may indicate any number between 0 and 1. (Values exactly 0 or 1 indicate without decimal places, i.e., indicating 0.0 or 1.0 is incorrect.)

RANDOM - (also default/unspecifying the subcommand) tightness between the two neighbouring clusters is defined by the macro randomly. Some neighbouring clusters may appear mutually tight, while other neighbouring clusters – quite separated.

**SIZING**

Parameter defining physical different-sizedness of clusters, inequality of their st. deviations, or “radii”. Specify one of the following:

*Number from 0 to 1* - 0 creates equally sized clusters; 1 creates strongly unequal-sized clusters: the radius of the biggest cluster to the radius of the smallest cluster ratio will be 3:1. You may indicate any number between 0 and 1. (Values exactly 0 or 1 indicate without decimal places, i.e., indicating 0.0 or 1.0 is incorrect.)

RANDOM - (also default/unspecifying the subcommand) the ratio of sizes between the biggest and the smallest clusters will be set by the macro as a random number between 1 and 3.

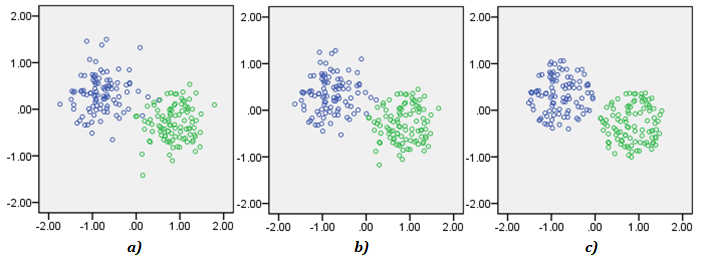
**FLAT**

This subcommand does not act with ELONG=EXTEND. Parameter defining kurtosis in the distribution of within-cluster points (see picture). By default and with FLAT=0, distribution in clusters is normal. In order to make distribution more platykurtic, specify one of the following:

*Number from 0 to 1* - degree of blunt-topness, equal for all the clusters. The closer the number to 1, the stronger is blunt-topness. Under FLAT=1 the distribution reaches to be uniform, that is, clusters are hyperballs.

RANDOM - degree of blunt-topness, randomly varying from cluster to cluster. Parameter from 0 to 1 will be set by the macro for each cluster, as a random number.

Within-cluster multivariate variance in a flattened cluster will be approximately the same as with FLAT=0, and in a large series of simulations clusters generated under different value of FLAT (and equal in all other respect) may be considered having the same variance.



1. FLAT=0 (normal distribution)
2. FLAT=0.5
3. FLAT=1 (uniform ball - disk)

**ELONG**

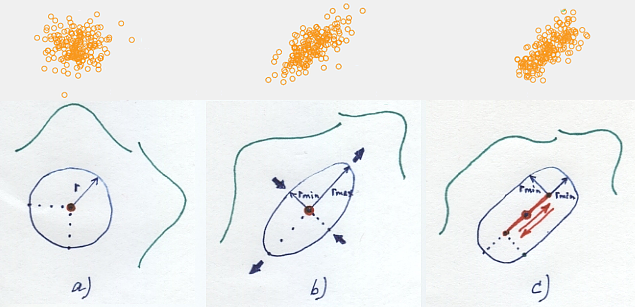
By default/unspecifying of this subcommand, clusters are created round, spherical. ELONG (the subcommand is permitted if *p*>1) allows to make clusters oblong. Clusters get oblong only in some one direction of space; remaining spherical in the other dimensions. In other words, each cluster presents itself as a cloud, ideally having single dominating principal axis. Indicate the needed kind of oblong clusters:

STRETCH *min max* - stretched clusters. Such a cluster comes out of a round cluster by stretching its variance in one orientation (bipolar direction) of space, squeezing it by all other dimensions of space (and if the round cluster was with normal distributions of points, it will retain normal distribution). Density of points per unit metric is lessened along one orientation and raised along the rest orientations.

EXTEND *min max* - extended clusters. Such a cluster has platykurtic distribution in one orientation – in which it is oblong – and normal one in other orientations, orthogonal to it. The cluster is more compact and denser, than the stretched cluster. Extended cluster is generated the same way the round cluster is, but during generating of points the centre randomly runs all the time on a cut of straight line, rather than stands still in one place.

Parameters *min* and *max* are two numbers in the range 1 to 4. They define the degree of cluster oblongness; the greater the number the more oblong is a cluster (1 is equivalent to round cluster). If you specify two different numbers, clusters will have unequal degree of oblongness, and for each cluster the degree of oblongness will be a random value between min and max. If you specify two equal numbers, all the clusters will have the same degree of oblongness. It doesn’t matter in what order you specify the numbers – min max or max min.

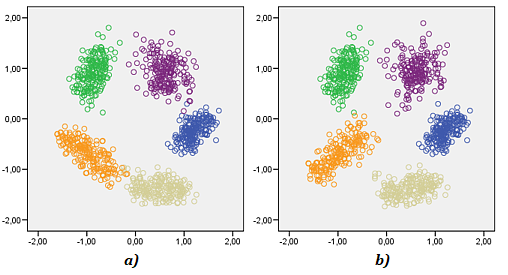
The macro generates oblong cluster approximately with the same magnitude of overall multivariate variance that the cluster has when it is round, – if the settings of generation are all the same (i.e., all other specifications except ELONG s/c are the same and the random number seed the user set is the same). The three kinds of clusters are compared on the picture below.



1. Round normally distributed cluster. Cluster points are generated under standard deviation r in all directions.
2. Stretched cluster. Similar to round cluster, but is stretched in one bipolar direction and squeezed in the other one, i.e., cluster points are generated under st. deviation rmax and rmin, correspondingly.
3. Extended cluster. At generating the points, it has mobile centre finding itself in a random location on a straight line along one bipolar direction. Сluster points are generated under st. deviation rmin in all sides, similar to how it is in a round cluster.

**ROTAT**

This subcommand is in effect only if ELONG is specified. Elongated clusters are oblong along one random orientation in space; else speaking, they are randomly rotated in space by their 1st principal axis. If you want this rotation to be perfectly random, choose ROTAT=FREE (also default/unspecifying the subcommand). Then different clusters may occur to be rotated independently of each other and in any way. If you specify ROTAT=RESTR, then the 1st principal axes of all the clusters will be co-oriented to this or that degree, so all the clusters will be more or less positively correlated clouds (see the picture). And if you are denying rotation at all, specify ROTAT=NONE. The 1st principal axis of every cluster will be parallel with the variable *V1* axis. (You can perform rotation later by !KO\_ROTCLU macro.)



1. ROTAT=FREE. Clusters are oblong in random direction.
2. ROTAT=RESTR. All clusters are to a degree co-oriented.

**CLEAR**

Clusters are always generated first as fuzzy – it is clusters which can intersect, overlap, if they are located tight enough to each other. S/c CLEAR allows you to make such clusters more clear, demarkating them in the region of intersection. Through this, some points may change their cluster membership. Choose the method:

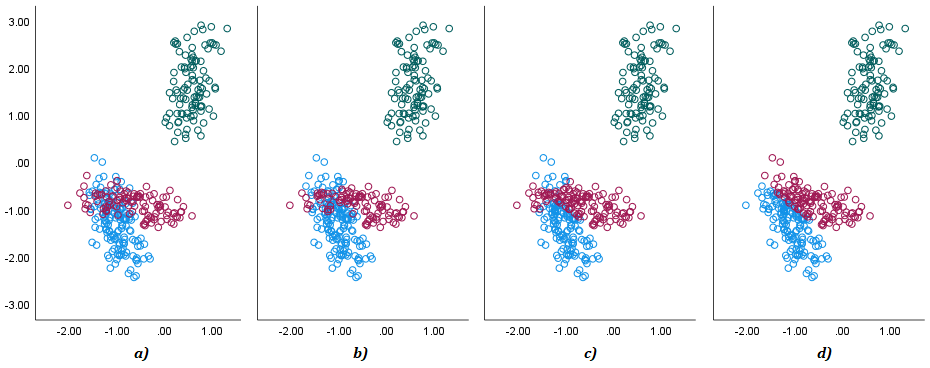
KNN - classification by K-nearest neighbours method. For each point, its nearest neighbours (by the Euclidean distance) are determined, and their cluster membership is asked. The point will be assigned to the cluster that prevails among the said neighbours. This method demarcates clusters gently, incompletely. SPSS command KNN is used.

LDA - classification by discriminants method. Performed is linear discriminant analysis, with extraction of all the discriminants from the data and then classification of points with them, dictated by their separate covariance matrices. To put it differently, this is the assignment of points to the clusters nearest to the points according to Mahalanobis distances in the space of the discriminants. This method demarcates clusters strongly, but in case of considerably elongated cross-intersecting clusters the latter may preserve the fact of intersection. SPSS command DISCRIMINANT is used.

KMC - classification by К-means method. K-means clustering without iterations is done: only assigning of points to the nearest to them centres is performed, the centers used being the initial ones – under which points were generated. This is the most radical method of demarcation, but it may not be suited for considerably oblong clusters, because it operates with Euclidean distances (which do not pay attention to possible covariational heterogeneity among clusters). SPSS command QUICK CLUSTER is used.

For KNN method, you can specify the number of nearest neighbours. Add subcommand /NEIGH=*number*. By default, *number* equals 3. For LDA method, you can request prior probability to reflect cluster’s populatedness with points. Add subcommand /PRIORS=SIZE. By default, prior probability is equal for clusters.

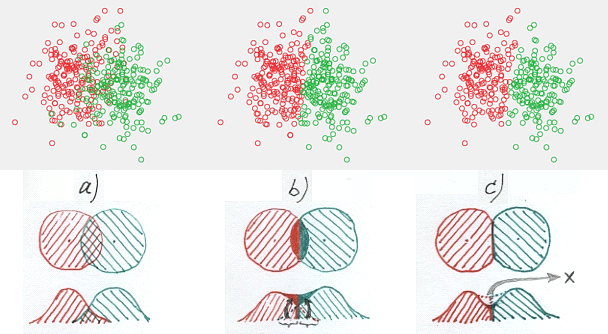
Methods LDA and KMC may affect cluster shape considerably. If two round clusters overlapped strongly, they, becoming clear, will appear semicircle. On the other hand, if clusters were created elongated, they may, with KMC, sometimes “catch from the side” alien points and thus become rounder (so is the nature of K-means method, which is isotropic in all directions and opts to leave approximately round clusters after it).



1. Fuzzy clusters, mixture (CLEAR is not specified).
2. CLEAR=KNN. Gentle demarcation.
3. CLEAR=LDA. Strong demarcation, but the intersection remained (due to cluster oblongness).
4. CLEAR=KMC. Clusters are demarcated to adjacency.

**WEED**

The subcommand acts only with CLEAR subcommand specified, and orders the “weeding” of the boundary zones of demarcated clusters from the “excessive” density of points. Indicate a number between 0 and 1. WEED=0 (also default/unspecifying the subcommand) means not to weed after demarcating of clusters. WEED=1 means to do a full weeding. (Values exactly 0 or 1 indicate without decimal places, i.e., indicating 0.0 or 1.0 is incorrect.) The essence of no weeding and full weeding is shown in the picture below. Weeding removes points that have changed their cluster membership due to s/c CLEAR.



1. Intersecting (fuzzy) clusters, mixture.
2. Demarcated (clear) clusters, without weeding. The points having come to the given cluster from the other one increase the density of the cluster in the boundary zone. Two clusters, although clear, still represent together a mixed distribution almost lacking bimodality.
3. Demarcated (clear) clusters, with full weedeing. The points having come to the given cluster from the other one were all deleted. The two clusters together represent obviously bimodal distribution.

If you specify for WEED a number *between* 0 and 1, a partial weeding will be done, whereby from each cluster there will be removed only part of the points having come to it from other clusters, – the proportion equal to the specified value WEED. Candidate points for deletion will be selected randomly from them, however, basically those points receive greater chance to be deleted that stand farther from the initial centre of the given cluster in the direction of the initial centre of the cluster from which they came.

Any WEED >0 means that if any clusters intersected before their demarcation, some points from them will be checked out. As a result, populatedness of clusters with points will often be less than expected according subcommand N specidied by the user. Therefore, it obviously makes sense for the user to overstate (give “with a margin”) specification of N s/c in case CLEAR plus WEED close to 1 are planned, while SPACING is close to 0. Sometimes, with 0<WEED<1, some clusters may end up with more points than they are scheduled according to N.

***Special regimes***

Since the macro does not have input data, it is indifferent to things like weightedness, splitness or filteredness of the working dataset.

***Algorithm***

Алгоритм придуман автором макроса.

k - число кластеров; p - число переменных (размерность пространства).

Этап I. Порождение инициальных центров кластеров.

-Если заказано создать центры случайными, порождаем их как k точек из равномерного распределения.

-Если заказано создать центры примерно равноразделенными, то порождаем 100\*k\*p случайных точек, причем

половина из равномерного (прямоугольного) распределения, а половина из нормального распределения; оба

распределения должны иметь одни и те же параметры средней и ст. отклонения. Это облако точек подвергаем

кластеризации K-СРЕДНИХ с дефолтными настройками и требованием выделить k кластеров. Центры последних

и есть искомые инициальные примерно равнораспределенные центры. (Создание смеси точек из прямоугольного

и нормального распределений потребовалось потому, что кластеризация только лишь квадратного облака

имеет тенденцию дать центры кластеров, расположенные более или менее крестообразно, а кластеризация

только лишь круглого, вроде нормального, облака имеет тенденцию дать центры, расположеннные

кольцеобразно. Нам желательно уравновесить эти две тенденции.)

Этап II. Задание размеров кластеров и степени тесноты между соседними кластерами.

r - размер, или "радиус" кластера; это величина ст. отклонения в нем.

m - отношение r\_max/r\_min (т.е. отношение r между самым большим и самым малым из порождаемых кластеров),

Значение m задается один раз пользователем либо генерируется случайно, в диапазоне от 1 до 3.

q - теснота соприлегания между двумя соседними кластерами, именно, между каждыми двумя кластерами, один

из к-рых является ближайшим соседом второго. q измеряется в количестве r, укладывающихся между

центрами 2-х кластеров. Желательное (не во всех соседних кластерах в итоге выдерживаемое) значение q

задается, в диапазоне от 0.4 до 8.4, либо единожды пользователем, либо заново для каждых двух

соседних кластеров случайным порождением.

Цель этапа - вычислить такие r кластерам, к-рые учитывали бы заданные параметры m и q.

1. Вычислим евклидовы расстояния между всеми k инициальных центров.

2. Для каждого центра, найдем его ближайшего соседа, т.е. другой центр, отделенный от него наименьшим

расстоянием (расстояние до ближайшего соседа назовем d). Создадим табличку: номер центра п/п, номер

его ближ. соседа, d; вот пример таблички:

Центр Бл.Сосед D

1 4 4.56

2 6 2.97

3 1 5.48

... ... ...

k 2 6.55

3. Вычислим среднегеометрическую d (назовем ее d\_mean).

4. -Если m=1, т.е. все кластеры должны быть одного размера, то их r= d\_mean/q. Конец этапа.

-Если m>1, то r\_min= 2\*d\_mean/[q\*(m+1)] и r\_max= r\_min\*m. (r\_min и r\_max, посчитанные впервые,

запомним как r\_min1 и r\_max1.) Отдадим r\_max центру, у к-рого d максимально (наибольшее значение

в столбце D таблички). Отдадим r\_min центру, у к-рого d минимально (наименшее значение в столбце D).

(Таких центров, с мин. d, в действительности будет два, - неважно, какому из них отдать r\_min.)

5. Пробегаем список центров (1-й столбец таблички). Если центр i еще не имеет r, а центр-его ближайший

сосед (j) имеет, вычислим r для центра i: r(i)= 2\*d/q - r(j), где d - расст. между i и его ближ.

соседом j. Вычисленное r(i) нельзя выпустить за пределы от r\_min1 до r\_max1, поэтому, если

r(i)<r\_min1, то пусть r(i)= r\_min1; если r(i)>r\_max1, то пусть r(i)= r\_max1.

6. Повторяем пробежки (п. 5) и вычисление r для центров, пока не обнаружится, что после очередной

пробежки число центров с вычисленным r не изменилось.

-Если число центров с r =k, конец этапа.

-Если число центров с r <k, тогда удалим из таблички все ряды (т.е. центры), для которых r вычислен,

и идем опять к п. 3. В п. 4, при вычислении новых r\_min и r\_max надо будет учесть, что они не имеют

права выйти за пределы от r\_min1 до r\_max1, поэтому если r\_min<r\_min1, пусть r\_min= r\_min1;

если r\_max>r\_max1, пусть r\_max= r\_max1.

Этап III. Порождение кластеров.

-Порождаем каждый кластер как случ. данные из стандартного нормального распредедения. Если кластер должен

быть более туповершинным, чем нормальный, умножаем координаты каждой точки на коэффициент сжатия, равный

1 + flat \* [CDF.CHISQUARE(D,df=p)^(1/p) \* sqrt(p+2) / sqrt(D) - 1], где D – квадратное отклонение точки

от центра кластера; flat – параметр (от 0 до 1) заданный пользователем или выбранный случайно.

-Если кластеры должны быть круглыми, то:

1. Порождаем каждый кластер как случ. данные из стандартного нормального распредедения.

-Если кластер должен быть более туповершинным, чем нормальный, умножаем координаты каждой точки на

коэффициент сжатия, равный 1 + flat \* [CDF.CHISQUARE(D,df=p)^(1/p) \* sqrt(p+2) / sqrt(D) - 1], где

D – квадратное отклонение точки от центра кластера; flat – параметр (от 0 до 1) заданный

пользователем или выбранный случайно.

2. Придаем кластеру ст. отклонение (умножив его данные на r) и центр (прибавляем координаты инициального

центра.

-Если кластеры должны быть вытянутыми в одном случайном направлении, то:

1. Делаем то же, что пункт 1 для круглых кластеров.

2. Для каждого кластера имеем (задано или порождаем) коэф-т продолговатости u (число от 1 до 4) и

порождаем матрицу случайного ортогонального поворота rm. Последняя есть ортобазис Грама-Шмидта

матрицы размером pxp, состоящей из случ. чисел из нормального распределения

(причем если ROTAT=RESTR, 1-й столбец этой матрицы должен содержать только положит. числа,

а полученный из матрицы ортобазис rm транспонируем).

3. Придаем кластеру ст. отклонение, равное r\_уменьшенное\*u, для первой переменной (V1), и равное

r\_уменьшенное, для остальных переменных; r\_уменьшенное= sqrt[p\*r^2/(p-1+u^2)].

4. Данные каждого кластера поворачиваем в пространстве, умножая на заготовленную для них случайную

матрицу поворота rm (пункт 2).

-Если кластеры должны быть продленными в одном случайном направлении, то:

1. Для каждого кластера имеем (задано или порождаем) коэф-т продолговатости u (число от 1 до 4) и

порождаем координаты некоторой точки ("направляющая точка"), отклоняющейся от центра кластера

в случайном направлении пространства (причем если ROTAT=RESTR, то все ее координаты

делаем положительными).

2. Требуемое расстояние dcd направляющей точки до центра вычислим как (e - основание нат. логарифма):

dcd= e^(.4+1.42/u)\*r\_уменьшенное\*(u-1), где r\_уменьшенное= см. ф-лу выше. Пропорционально изменим

координаты точки так, чтобы привести ее отстояние от центра кластера к величине dcd.

3. Порождаем каждый кластер как случ. данные из нормального распределения с параметрами: средняя=

координата бегающего центра; ст. отклонение= r\_уменьшенное. Бегающий центр это точка, находящаяся

в случайном месте на отрезке от направляющей точки до точки, зеркально симметричной ей относительно

инициального центра кластера. Бегающий центр изменяем при порождении каждого очередного данного.

Если кластеры заказано создать нечеткими - конец работы.

(Нечеткие кластеры могут перекрываться своими перифериями, если кластеры достаточно тесны друг к другу.)

Этап IV. Делание кластеров четкими и прополка их пограничных зон.

Этот этап может быть излишен, если кластеры заказано создать широко раздвинутыми, т.к. тогда они

наверняка не перекрываются.

1. Дадим (i) кластерному анализу K-средних (команда QUICK CLUSTER) либо

(ii) линейному дискриминантному анализу (команда DISCRIMINANT) либо (iii) анализу K-ближайших соседей

(команда KNN) отклассифицировать порожденные точки (наблюдения) к тем кластерам, к которым они ближе.

(Примечание 1: QUICK CLUSTER делается в режиме Classify only, т.е. без итераций, и инициальные

кластерные центры ждя кдассификации – это инициальные центры порождения точек.

Примечание 2: DISCRIMINANT пускается с п/к CLASSIFY=SEPARATE, т.е. классифицируем без допущения, что

ковариации в кластерах сходны, и классификация делается каноническими дискриминантами.) Если прополка

пограничных зон (участков, где кластеры пересекались) не требуется – конец работы.

2. Прополка пограничных зон означает удаление части порожденных наблюдений: численность в кластерах

будет меньше, чем заказано пользователем в подкоманде N.

-Если требуется полная прополка, то удаляются все точки, кластерная принадлежность к-рых изменилась

при классификации в п. 1.

-Если требуется частичная прополка (требуется удалить из кластера долю 0<weed<1 точек, пришедших в

него из других кластеров в результате классификации п. 1):

2.1. Для каждой точки, пришедшей в "наш" кластер из данного "чужого" кластера, вычислим расстояние

точки до инициального центра нашего кластера (d1), расстояние ее до инициального центра чужого

кластера (d2), а также расстояние между инициальными центрами нашего и чужого кластеров (d3).

2.2. Имея эти 3 дистанции, вычислим длину проекции точки на линию, соединяющую центры нашего и

чужого кластеров: (d1^2+d3^2-d2^2)/sqrt(d3^2)/2. Умножим эту величину на случайное положит.

число. Проделаем описанное для всех точек, пришедших в наш кластер из данного чужого кластера

(случайное число каждый раз порождая новое).

2.3. Удалим из всех точек, пришедших в наш кластер из данного чужого, долю=weed точек с наибольшими

получившимися таким образом значениями.

Т.е. мы удалили из кластера нужную долю (weed) точек, бывших прежде "чужими", выбрав из них

случайно, однако с тем, чтобы точки, лежащие дальше от центра нашего кластера в направлении к

центру своего бывшего кластера, имели базово больший шанс быть удаленными.

# MACRO !KO\_ROTCLU: RANDOM ROTATION OF (CLUSTER) DATA

Version 2, Dec 2010 (Version 1, Nov 2010). Tested on SPSS Statistics 13, 15, 17.

!KO\_rotclu vars= *v1 v2* /\*Variables representing points coordinates; minimum two, name-by-name

/\*and/or via to

/rotat= FREE\_PAR /\*Demanded rotation: by any angle, own for each cluster (FREE, default);

/\*by any angle, one for all the clusters (FREE\_ANG);

/\*by any angle, making the clusters parallel by their pr axes (FREE\_PAR);

/\*into the positive and negative space quadrants, the angle own for each cluster (RESTR);

/\*into the positive and negative space quadrants, clusters parallel by their

/\*pr axes (RESTR\_PAR);

/\*remove rotatedness (UNROT), it is rotation into pr components

/pc= DIRECT /\*The nuance of preliminary rotation into pr components: allow counter-direction

/\*of pr axes (ORIENT, default); allow only co-direction (DIRECT).

Minimal specification VARS.

The macro takes data consisting of one or more groups, or “clusters”, of cases (those may be really clusters or totally differently formed groups) and rotates each cluster in the space axes-variables about its centroid (multivariate mean). The rotation is by a random angle depending on the sown random number seed. There are also options to eliminate rotation, give clusters the same rotatedness or co-directedness, and so on.

Input is multivariate data of the working dataset, where there must be present the numeric grouping variable *CLUSTER\_* with group (cluster) codes. Codes may be any numbers. The dataset *must be sorted* by ascending of values of that variable. The variable may be a constant (which means your data consist just of one cluster). The macro outputs the rotated data as a new working dataset. In the input data (including the *CLUSTER\_* variable) missings are unacceptable.

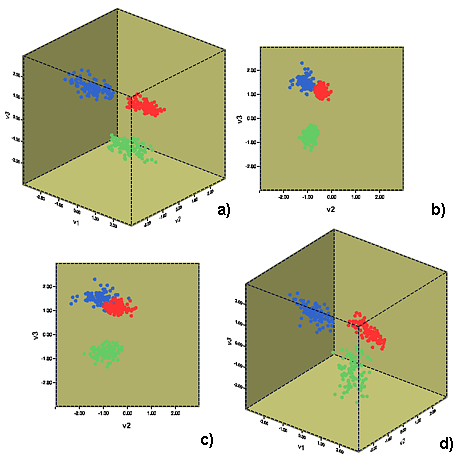
ПРИМЕР 1. Создание эллипсоидных повернутых кластеров. Макрос !KO\_GENCLU (см. выше) может порождать продолговатые кластеры, однако это сфероидные кластеры: они продолговаты только в одном направлении многомерного пространства, оставаясь в остальном сферическими. Другими словами, в каждый кластер заложена исходно однофакторная модель (в терминологии факторного анализа), сколько бы переменных ни было создано. Эллипсоидные (следовательно многофакторные) облака-кластеры можно сделать, модифицировав вручную кластеры, порожденные в !KO\_GENCLU, а затем случайно повернув их в пространстве макросом !KO\_ROTCLU.

!KO\_genclu k= 3 /p= 3 /n= 100 /spacing= .8 /sizing= .4 /elong= EXTEND 3 3 /rotat= NONE.

aggregate /outfile= \* mode= addvariables /break= cluster\_ /v2\_mean= mean(v2).

compute v2= (v2-v2\_mean)\*2+v2\_mean.

!KO\_rotclu vars= v1 v2 v3.



* !KO\_GENCLU породил 3 трехмерных кластера. Кластеры в данном случае продолговатые (по типу продленных). Они продолговаты все параллельно друг другу и оси V1, поскольку поворот не заказывался (ROTAT=NONE), - см. рис. a. Кластеры сфероидные: в проекции V2-V3 они круглые (рис. b).
* Исследователь решил увеличить в 2 раза стандартное отклонение во всех кластерах по переменной V2. Для этого он выяснил средние кластеров по этой переменной (команда AGGREGATE), перевел ее данные в отклонения от средний, умножил на 2 и вернул среднюю обратно (команда COMPUTE). Теперь кластеры эллипсоидные: они вытянуты не только вдоль V1, но немного и вдоль V2 (рис. c).
* Наконец, исследователь придал эллипсоидным кластерам случайную повернутость в пространстве макросом !KO\_ROTCLU (рис. d).

***Подкоманды***

**VARS**

Укажите минимум две числовые переменные, являющие собой собственно данные – координаты наблюдений. Список можно писать поименно и/или ч-з to.

**ROTAT**

Задайте особенности поворота. Следующие три варианта не накладывают ограничения на итоговую ориентацию кластеров в пространстве: угол их повернутости может оказаться любым случайным.

FREE - (тж. по умолчанию/незаданию) Кластеры поворачиваются независимо друг от друга, каждый кластер на свой угол.

FREE\_ANG - Поворот всех кластеров на один и тот же угол. Кластеры, следовательно, сохраняют исходную повернутость относительно *друг друга*.

FREE\_PAR - Поворот кластеров, который делает их облака параллельными друг другу: 1-е главные оси кластеров параллельны, 2-е главные оси их параллельны, и т.д.

Следующие два варианта тоже делают случайный поворот, но накладывают такое ограничение на итоговую повернутость, чтобы первая главная ось каждого кластера была ориентирована непременно между «углом чисто отрицательных» и «углом чисто положительных» координат пространства. Если кластеры продолговатые, они, стало быть, все станут положительно коррелированными облаками.

RESTR - Кластеры поворачиваются независимо друг от друга, каждый кластер на свой угол.

RESTR\_PAR - Поворот кластеров, который делает их облака параллельными друг другу: 1-е главные оси кластеров параллельны, 2-е главные оси их параллельны, и т.д.

Следующий вариант отменяют повернутость входящих кластеров. Это замена данных на их главные компоненты: 1-я главная ось каждого кластера станет параллельна 1-й оси пространства (т.е. 1-й переменной), 2-я главная ось – 2-й оси пространства (2-й переменной), и т.д.

UNROT - Поворот кластеров, который делает их главные оси параллельными друг другу и параллельными осям пространства.

Если данные состоят из единственного кластера, то FREE и FREE\_ANG дадут один и тот же результат, RESTR и RESTR\_PAR дадут один и тот же результат (при одном итом же зерне случайных чисел, разумеется).

**PC**

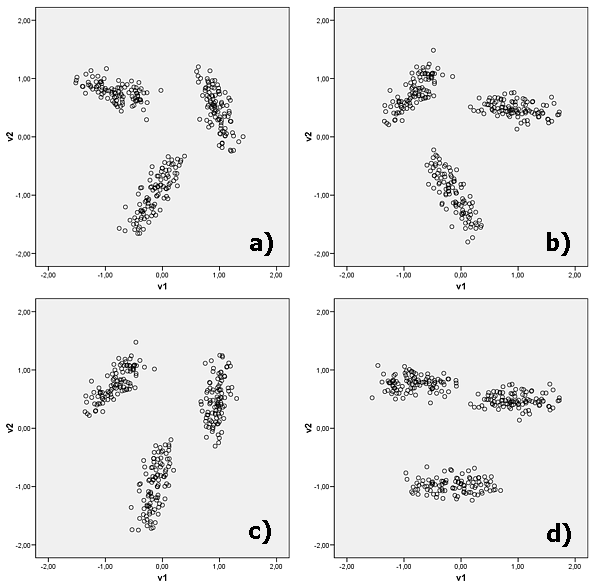
Подкоманда не действует при ROTAT=FREE или ROTAT=FREE\_ANG. В остальных случаях перед приданием кластерам окончательной повернутости с каждым из них проводится анализ главных компонент: исходные данные заменяются на главные компоненты. П/к PC задает нюансы этого.

ORIENT - (тж по умолчанию/незаданию) Углы поворота переменных (осей пространства) в главные оси облака могут быть от 0° до 180°. Если угол поворота n-й переменной в n-ю главную ось у одного кластера окажется меньше 90°, а у другого больше 90°, значит кластеры станут смотреть в одну сторону исходно противоположными краями. Словом, соответственные главные оси кластеров ориентируются параллельно, но не обязательно сонаправлено.

DIRECT - Углы поворота переменных в главные оси облака могут быть от 0° до 90°, для всех кластеров, поэтому кластеры не смогут повернуться друг к другу исходно противоположными краями. Словом, соответственные главные оси кластеров не просто параллельны, но сонаправлены.

Сочетание ROTAT=UNROT с PC=DIRECT – единственное задание, результаты при котором не зависят от случайных чисел.

На рисунках ниже – иллюстрация некоторых поворотов.



* a) Входящие данные, т.е. до поворота макросом.
* b) ROTAT=FREE\_ANG. Поворот всех кластеров на один и тот же случайный угол (здесь примерно на 50° против часовой стрелки).
* c) ROTAT=RESTR + PC=DIRECT. Ограниченный поворот: все кластеры ориентированы своей 1-й главной осью между левым нижним и правым верхним углами пространства. Обратите внимание, что те полюса всех кластеров, которые смотрели во входящих данных вправо, все по-прежнему смотрят вправо (т.к. PC=DIRECT).
* d) ROTAT=UNROT + PC=ORIENT. Устранение повернутости кластеров: их главные оси параллельны друг другу и осям пространства. Заметьте, что по сравнению с входящими данными не все кластеры сонаправлены, т.е. развернуты в одну сторону; левый верхний кластер повернулся более чем на 90°, тогда как другие – менее чем на 90°; это возможно при PC=ORIENT.

***Особые режимы***

Не используйте расщепление массива данных (SPLIT FILE) при работе с макросом. Макрос слушается фильтрации или отбора наблюдений (FILTER, SELECT IF, USE). Взвешивания не слушается.