

# Кластеризация многомерных данных

Общепринято, что какую-либо обработку статистических данных (усреднение, установление связей и т. д.) надо производить только в однородных группах наблюдений.

*И. Д. Мандель, «Кластерный анализ»*

# «Проклятие размерности»

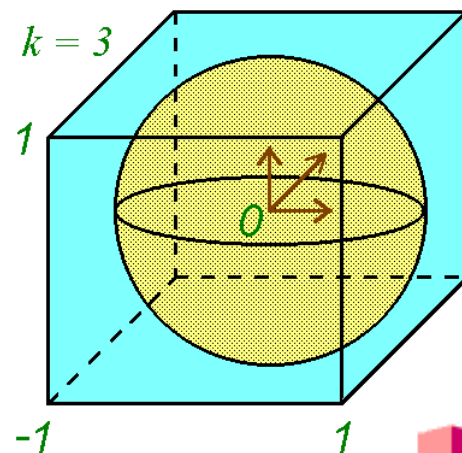
При использовании статистических методов для анализа таблицы данных могут возникнуть неожиданности, связанные с тем, что наша обычная геометрическая интуиция способна навести на неверное представление о  $k$ -мерных множествах.

Рассмотрим  $k$ -мерный шар  $\{\mathbf{x} : x_1^2 + \dots + x_k^2 \leq 1\}$ , вписанный в  $k$ -мерный куб  $\{\mathbf{x} : |x_j| \leq 1, j = 1, \dots, k\}$ .

Вероятность  $p_k$ , что выбранная случайно в кубе точка окажется внутри шара, равна отношению объема  $k$ -мерного шара радиуса  $r = 1$  к объему  $k$ -мерного куба со стороной 2. Очевидно,  $p_2 = \pi r^2 / 2^2 = \pi / 4 \approx 0,785$ ;  $p_3 = \frac{4}{3} \pi r^3 / 2^3 = \pi / 6 \approx 0,524$ .

**Вопрос.** Что подсказывает Вам интуиция о порядке этой вероятности при  $k = 10$ ?

Вероятность  $p_{10} \approx 2,5 \cdot 10^{-3}$ .  
В результате в среднем только одна из 400 случайных точек попадает внутрь вписанного десятимерного шара.



# Практическое задание 1

1) Напишите программу на языке R для моделирования  $n$ -кратного выбора точек наудачу из  $k$ -мерного куба  $[-1, 1]^k$  и подсчёта частоты попаданий этих точек в  $k$ -мерный шар радиуса 1 с центром в начале координат (используйте цикл `for`, функцию `runif`, команду `if` (условие) `{...}`)

Чтобы набрать код программы, щёлкните по кнопке с белым крестиком внутри зелёного кружка, находящейся в левом верхнем углу экрана и выберите в меню пункт R Script Ctrl+Shift+N. В появившемся окне введите код программы. Для выполнения программы выделите весь код и нажмите кнопку Run с зелёной стрелкой, находящуюся над окном с кодом

2) Задайте  $n = 1000$  и  $k = 10$

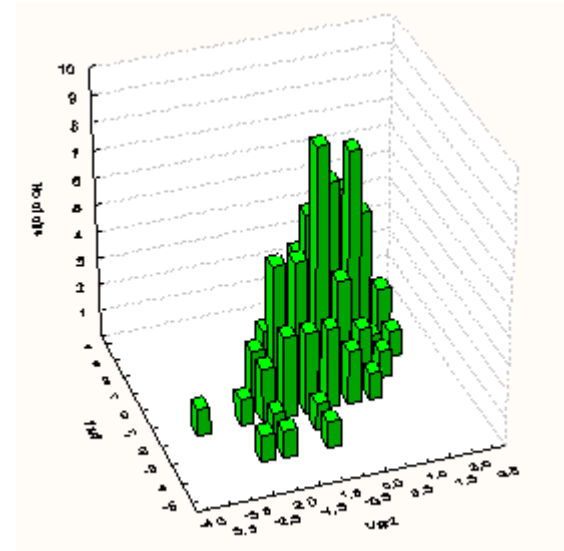
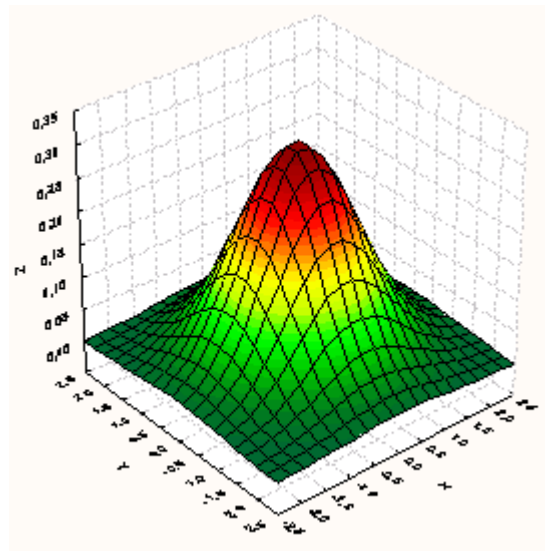
3) Запустите код несколько раз, обращая внимание на значение счётчика количества попаданий в шар

4) Вычислите приближённо вероятность, что количество точек, попавших в шар, окажется больше 8 (используйте формулу  $p_k = \pi^i 2^{-k} / i!$ , где  $k = 2i$ ,  $\pi$  для числа  $\pi$ , символ «^» для возведения в степень и функцию `factorial`)

5) Вычислите приближённо эту вероятность на основе теоремы Пуассона

6) Повторите пункты 3-5 для  $n = 10^6$  и  $k = 16$

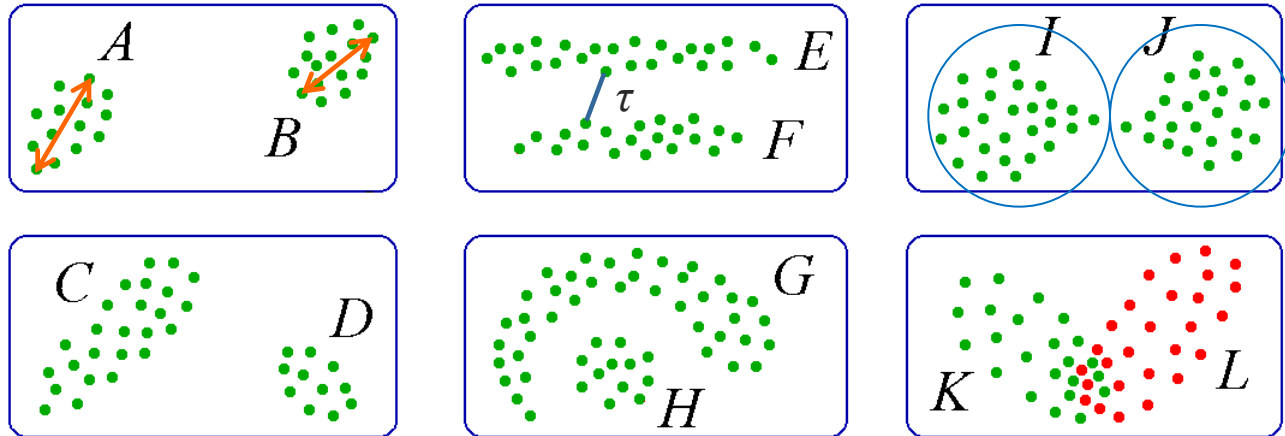
# Проблема проверки многомерной нормальности распределения



Допустим, что мы хотим проверить с помощью критерия хи-квадрат гипотезу согласия данных с некоторым многомерным законом распределения (например,  $k$ -мерным нормальным законом).

Предположим для простоты, что плотность распределения сосредоточена внутри  $k$ -мерного единичного куба. Возьмем  $h = 0,2$  в качестве длины ребра ячейки группировки. Желание иметь в каждой из ячеек не менее 5 наблюдений влечет неравенство  $nh^k \geq 5$ . Из него при  $k = 2$  находим, что  $n \geq 5 \times 5^2 = 125$ . Но при  $k = 10$  получаем, что  $n \geq 5 \times 5^{10} \approx \underline{50 \text{ миллионов!}}$  На практике количество наблюдений, имеющих у исследователя, редко достигает нескольких тысяч.

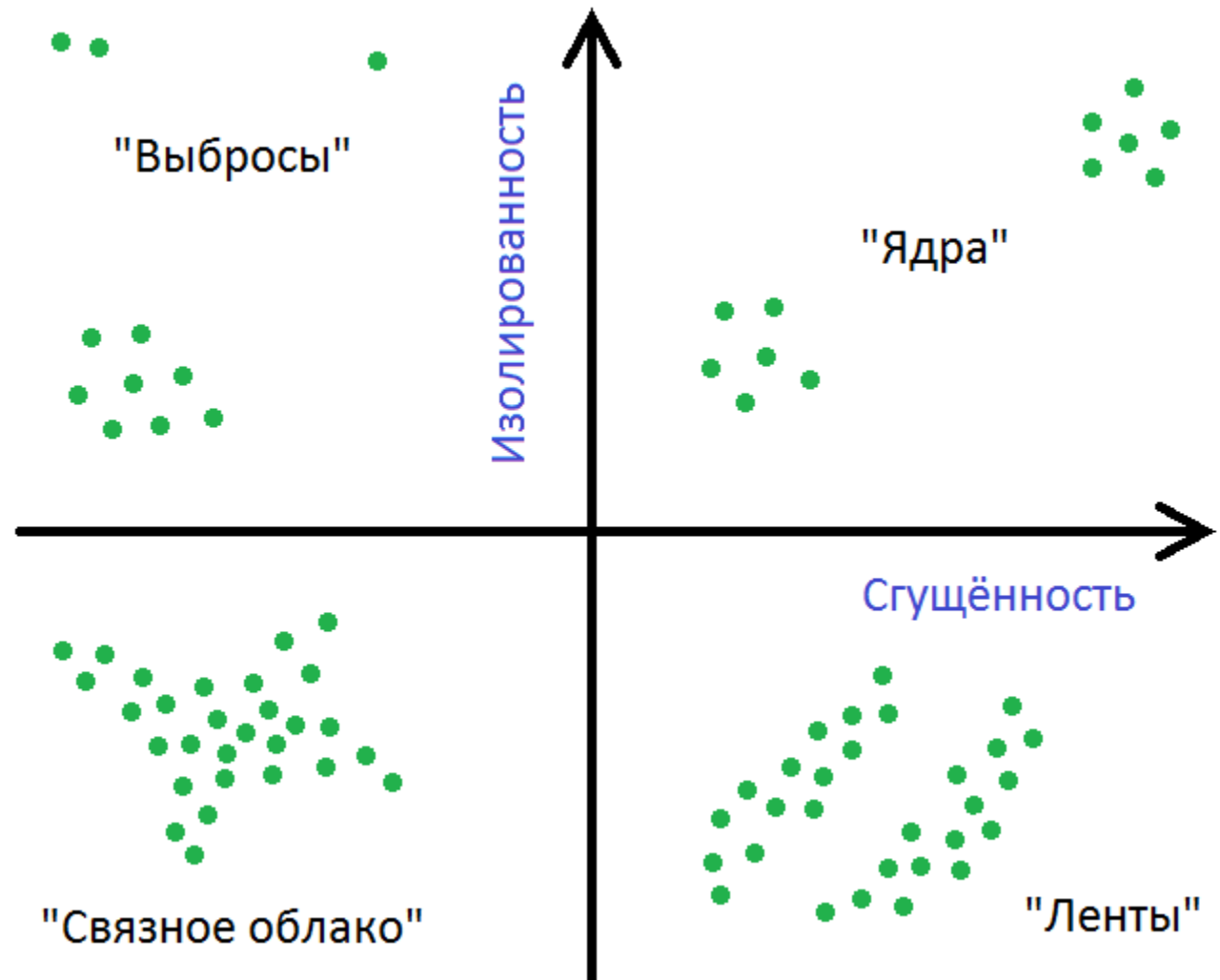
# Различные типы кластеров



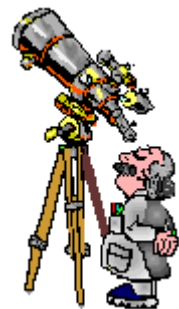
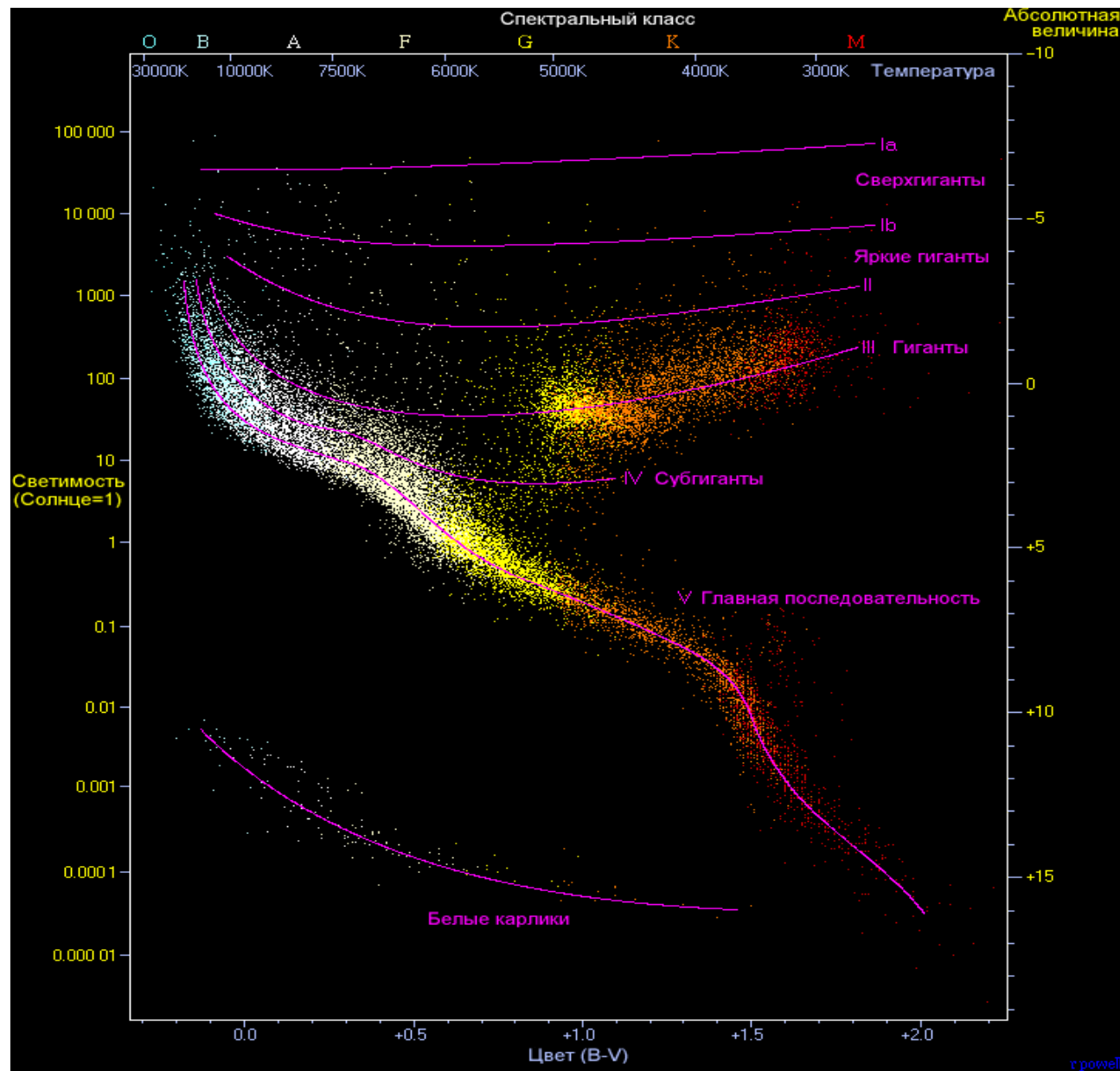
- **Кластер типа ядра:**  $A, B$  («чужие» — дальше диаметра)
- **Сгущение в среднем:**  $C, D$  (среднее межточечное расстояние)
- **Кластер типа ленты:**  $E, F, G, H$  («прыжок» длины  $\tau$ )
- **Кластер с центром:**  $I, J$  (внутри шара заданного радиуса)
- **Накладывающиеся разноцветные «облака» точек:**  $K, L$   
(такие множества точек будут рассматриваться позже при обсуждении темы «Классификация с обучением»)



# Сгущённость и изолированность



# Диаграмма «спектр – светимость»



# Ядерная оценка плотности (Розенблатта – Парзена)

This idea of an *average shifted histogram* or *ASH* density estimate is a useful motivation and is discussed in detail in Scott (1992). However, the commonest form of density estimation is a kernel density estimate of the form

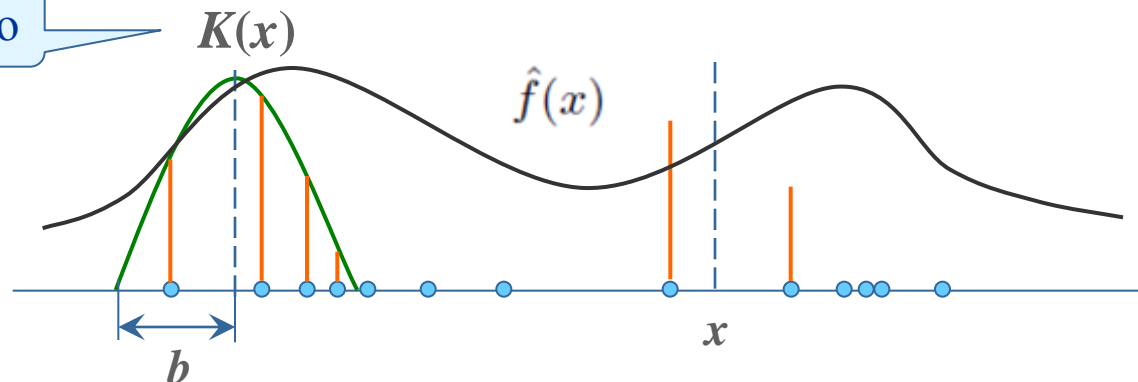
$$\hat{f}(x) = \frac{1}{nb} \sum_{j=1}^n K\left(\frac{x - x_j}{b}\right)$$

Ширина окна  
сглаживания

for a sample  $x_1, \dots, x_n$ , a fixed kernel  $K()$  and a bandwidth  $b$ ; the kernel is normally chosen to be a probability density function.

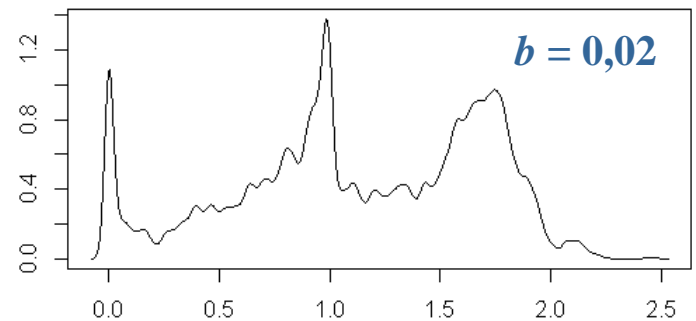
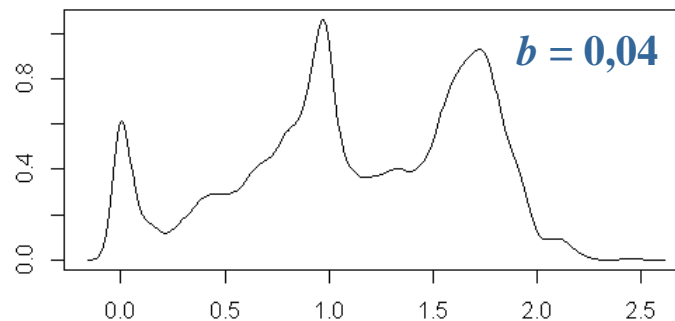
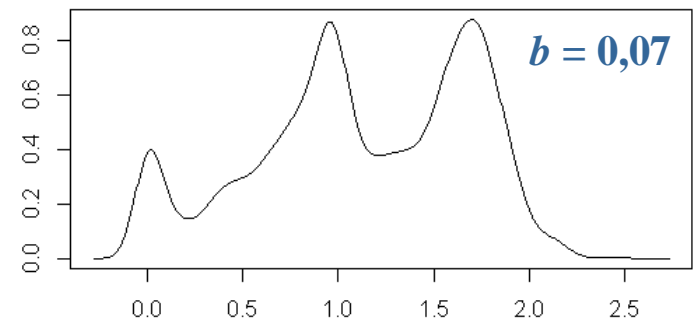
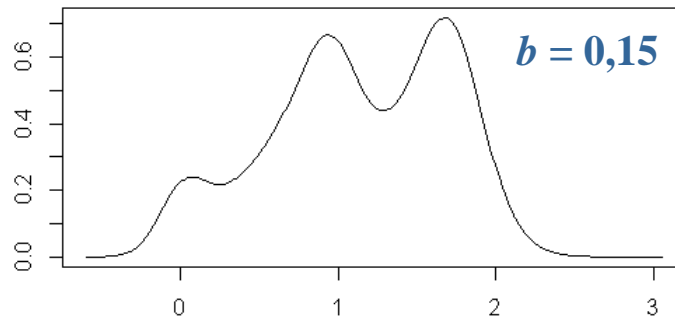
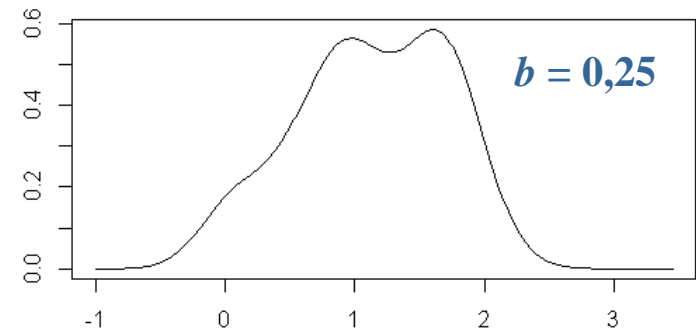
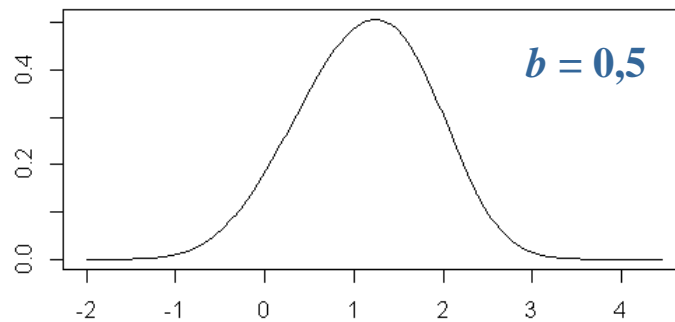
S-PLUS has a function `density`. The default kernel is the normal (argument `window="g"` for Gaussian), with alternatives `"rectangular"`, `"triangular"`.

Ядро

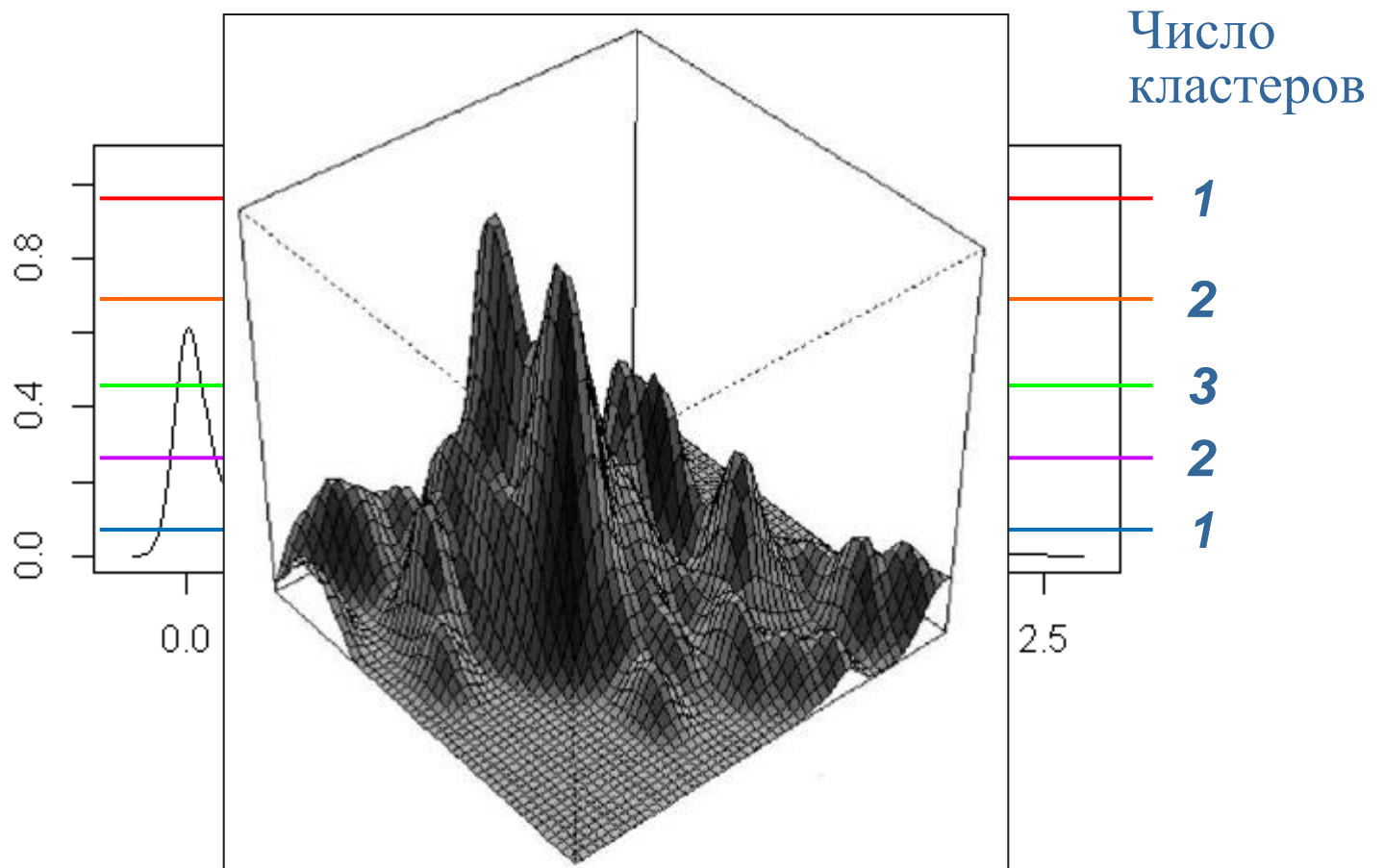




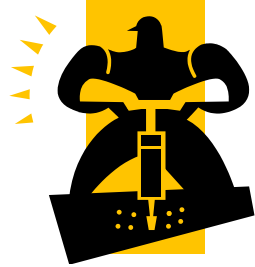
# Влияние ширины окна сглаживания на ядерную оценку



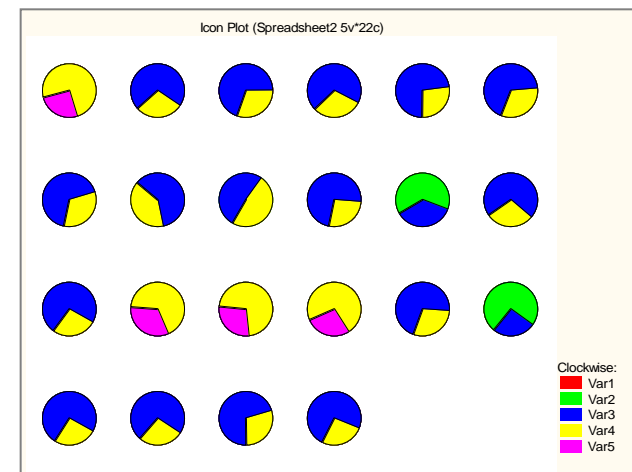
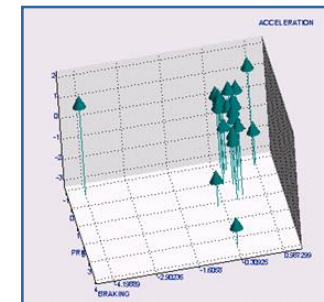
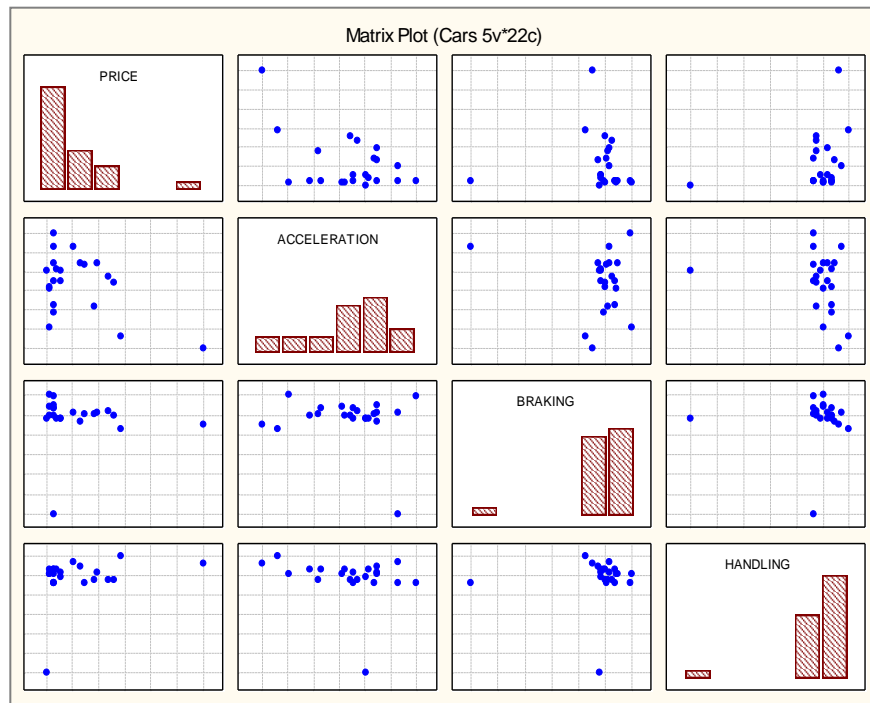
# Зависимость числа кластеров от выбора уровня сечения ядерной оценки плотности



# Разведочный анализ данных (Data Mining)



- **Matrix Plot** — матричная диаграмма рассеяния
- **3D XYZ Graph** — трёхмерная диаграмма рассеяния
- **Icon Plot** — диаграмма профилей



# Типы нормировки



**N1)**  $Z'_i = (Z_i - Z_{(1)}) / (Z_{(n)} - Z_{(1)})$ , где  $Z_{(1)} = \min\{Z_1, \dots, Z_n\}$ ,  $Z_{(n)} = \max\{Z_1, \dots, Z_n\}$ .

Заметим, что все величины  $Z'_i$  принадлежат отрезку  $[0, 1]$ .

**N2)**  $Z'_i = (Z_i - \bar{Z}) / S$ , где  $\bar{Z} = \frac{1}{n} \sum Z_i$ ,  $S^2 = \frac{1}{n-1} \sum (Z_i - \bar{Z})^2$ .  
Это преобразование обычно называют стандартизацией.

**N3)**  $Z'_i = (Z_i - MED) / MAD$ , где

*MED* — выборочная медиана,

*MAD* — (нормированная) медиана абсолютных отклонений от *MED*:

$$MAD = \frac{1}{\Phi^{-1}(3/4)} MED \{|Z_i - MED|, i = 1, \dots, n\},$$

где  $\Phi^{-1}(x)$  — функция, обратная к функции

распределения закона  $\mathcal{N}(0, 1)$ .

**N3** наименее подвержена влиянию «выбросов»

Median of Absolute Deviations

Этот множитель ( $\approx 1,483$ ) обеспечивает для выборки из распределения  $N(\mu, \sigma^2)$  сходимость оценки *MAD* к параметру  $\sigma$

Для выполнения нормировок **N1-N3** на языке R используйте функции `min`, `max`, `mean`, `std`, `median`, `mad`

# Кластеризация автомобильных марок

В файле CarChar.txt содержатся данные о 5 характеристиках автомобилей разных марок:

- 1 – *примерная цена* (Price),
- 2 – *ускорение* (Acceler, число секунд на разгон до 60 миль/час),
- 3 – *тормозное расстояние* (Braking, при скорости 80 миль/час до остановки автомобиля),
- 4 – *индекс способности удерживать дорогу* (Handling),
- 5 – *расход топлива* (Mileage, количество миль на галлон).





# Практическое задание 2

- 1) Импортируйте файл CarChar.txt (не перепутайте его с CarSales.txt) под именем d и подсчитайте число столбцов в таблице (функция ncol)
- 2) Несмотря на то, что признаки уже были нормированы с помощью стандартизации N2, примените к столбцам таблицы d нормировку N3, используя цикл, функции median и mad. Запишите результат в таблицу z
- 3) Выясните, как изменилось наибольшее значения признака PRICE
- 4) Найдите «выбросы» для всех нормированных признаков, построив boxplot для таблицы z без первого столбца, замените их на пропуски NA. Какие марки автомобилей выделяются и по каким признакам?
- 5) Постройте для таблицы z без первого столбца матричную диаграмму рассеяния (plot) (нажмите кнопку Zoom). Есть ли двумерные «выбросы»?
- 6) Постройте отдельную диаграмму рассеяния для признаков ACCELER и BRAKING, затем выполните команду text, задав аргумент labels=z[,1]
- 7) Примените к столбцам таблицы d нормировку N1, исключая пропуски при вычислении min и max (используйте функцию na.omit)
- 8) Постройте для марки «Buick» (строка 4) столбиковую диаграмму (barplot) и выясните, по каким именно признакам эта марка отличается от других (уберите названия марок, транспонируйте матрицу командой t)



# Эвристические алгоритмы

Подавляющая часть кластеризаций на практике проводится именно эвристическими методами. Это объясняется:

- 1) относительной простотой и содержательной ясностью таких алгоритмов;
- 2) возможностью вмешательства в их работу путем изменения одного или нескольких параметров, смысл которых обычно понятен;
- 3) невысокой трудоемкостью алгоритмов.

На практике наиболее часто используются метод « $K$ -средних» и иерархические процедуры.

# Метод «К-средних»

Число  
кластеров  $k$   
задаётся  
исследова-  
телем

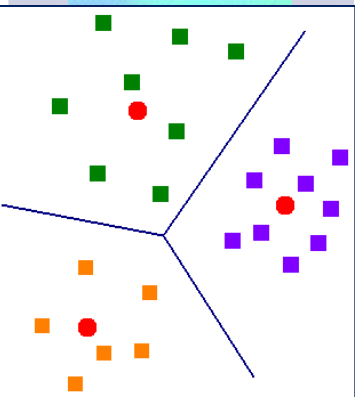
Метод «К-средних» предназначен для выделения кластеров типа «кластер с центром». Это название, ставшее популярным, введено Дж. Мак-Кином в 1967 г. Опишем один из вариантов метода.

Случайно выбираются  $k$  объектов — так называемых «эталонов». Затем каждый объект приписывается к ближайшему «эталону», тем самым объекты разбиваются на  $k$  групп. В качестве новых «эталонов» принимаются центры масс групп (считается, что каждый объект имеет массу 1). Другими словами, координаты новых «эталонов» определяются как средние арифметические координат объектов, входящих в соответствующую группу. После пересчёта объекты снова распределяются по ближайшим «эталонам» т. д. Критерием окончания алгоритма служит стабилизация центров масс всех групп. Итоговые группы считаются кластерами.

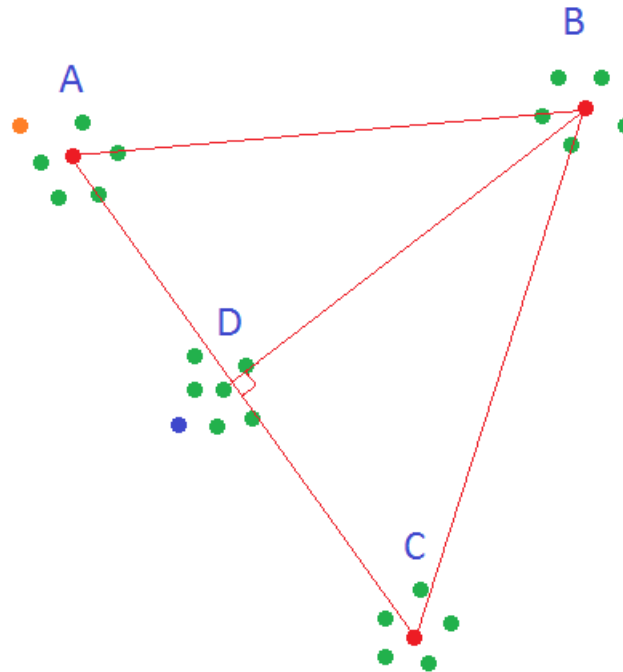
Обычно случайный выбор начальных «эталонов» повторяют несколько раз. Среди полученных кластеризаций наилучшую, как правило, выбирают по величине *межкластерной инерции*, определяемой как второе слагаемое в формуле

$$\sum_{i=1}^n |\mathbf{x}_i - \bar{\mathbf{x}}|^2 = \sum_{l=1}^k \sum_{\mathbf{x}_i \in S_l} |\mathbf{x}_i - \bar{\mathbf{x}}_l|^2 + \sum_{l=1}^k n_l |\bar{\mathbf{x}}_l - \bar{\mathbf{x}}|^2.$$

Здесь  $S_l$  — кластер с номером  $l$  ( $l = 1, 2, \dots, k$ ),  $n_l$  — число объектов в  $S_l$ .



# Выбор наиболее удалённых начальных «эталонов» может приводить к плохой кластеризации



Наиболее удалённой от трёх уже выбранных эталонов (красных точек) является не синяя точка из класса D, а оранжевая точка из класса A.

# Медиана Кемени

Для одномерной выборки легко убедиться, что минимум суммы расстояний до элементов выборки достигается на выборочной медиане *MED*. Это свойство можно обобщить на разбиения, если некоторым образом определить расстояние между ними.

Разбиение **A** можно описать матрицей из 0 и 1:  $a(i, j) = 1$ , если и только если объекты  $i$  и  $j$  входят в один класс. В 1959 году американский статистик Джон Кемени предложил использовать следующий способ вычисления расстояния между разбиениями:

$$d(\mathbf{A}, \mathbf{B}) = \sum_{i < j} |a(i, j) - b(i, j)|$$

Иначе его можно описать как количество пар объектов входящих в один класс для разбиения **A** (**B**) и входящих в разные классы для разбиения **B** (**A**).

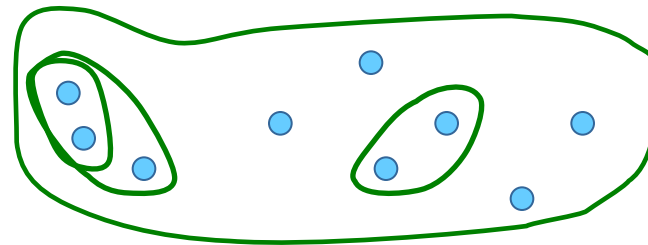
Кемени установил, что данное расстояние характеризуется рядом аксиоматических свойств (см. Орлов А. И. «Нечисловая статистика», 2004, с. 144). Разбиение, на котором достигается минимум суммы расстояний до других разбиений, называется их **медианой Кемени**.



# Практическое задание 3

- 1) Импортируйте файл `Simpl.txt` в RStudio под именем `s`
- 2) Командой `plot` и постройте для `s` матричную диаграмму и на её основе определите нижнюю границу для числа кластеров `k`
- 3) Примените к `s` процедуру `kmeans` с параметрами `k` и `nstart = 10`, реализующую метод *K-средних* со случайным 10-кратным выбором начальных «эталонов», и запишите результат процедуры в `r`
- 4) Постройте для `s` матричную диаграмму с раскраской точек в цвета, задаваемые номерами кластеров `r$cluster`
- 5) Вычислите долю разброса, приходящуюся на межкластерную инерцию относительно общей инерции:  $r\$betweenss/r\$totss$
- 6) Командой `dist` вычислите попарные расстояния между итоговыми «эталонами», координаты которых записаны по строкам в матрицу `r$centers`, и выведите эти расстояния на экран
- 7) На основе `r$centers` постройте *профили кластеров* — ломанные, соединяющие средние значения признаков в каждом кластере (примените команду `plot` с аргументом `type="l"`, затем команду `lines` в цикле)
- 8) Определите истинное число классов, увеличивая параметр `k` и повторяя пункты 3-7. Для найденной наилучшей кластеризации интерпретируйте кластеры с помощью профилей кластеров

# Иерархические процедуры



Опишем общую схему этих процедур.

Сначала каждый объект считается отдельным кластером. На первом шаге объединяются два ближайших объекта, тем самым создавая новый кластер.

Для нового кластера вычисляются *меры отдалённости* (см. след. слайд) от него до всех остальных кластеров.

Шаги процедуры повторяются до тех пор, пока все объекты не объединятся в один кластер.

Если прервать процедуру объединения за несколько шагов до полного объединения, то получим разбиение объектов на несколько кластеров. Наилучший момент прерывания можно определить визуально с помощью *дендрограммы* (от *déndron* (*греч.*) — дерево).

Примеры дендрограмм будут приведены ниже

# Типы иерархических процедур

**P1)** МЕТОД «БЛИЖНЕГО СОСЕДА»:  $\rho_{min} = \min_{x_i \in S_k, x_j \in S_l} d_{ij},$

**P2)** МЕТОД «ДАЛЬНОГО СОСЕДА»:  $\rho_{max} = \max_{x_i \in S_k, x_j \in S_l} d_{ij}.$

**P3)** МЕТОД СРЕДНЕЙ СВЯЗИ:  $\rho_{ave} = \frac{1}{n_k n_l} \sum_{x_i \in S_k} \sum_{x_j \in S_l} d_{ij}$

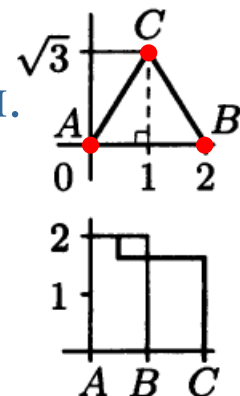
(здесь  $n_k$  и  $n_l$  — количества объектов в кластерах  $S_k$  и  $S_l$ ).

**P4)** МЕТОД ЦЕНТРОВ МАСС:  $\rho_{center} = |\bar{x}_k - \bar{x}_l|^2,$   
где  $\bar{x}_k$  и  $\bar{x}_l$  обозначают центры масс  $k$ -го и  $l$ -го классов.

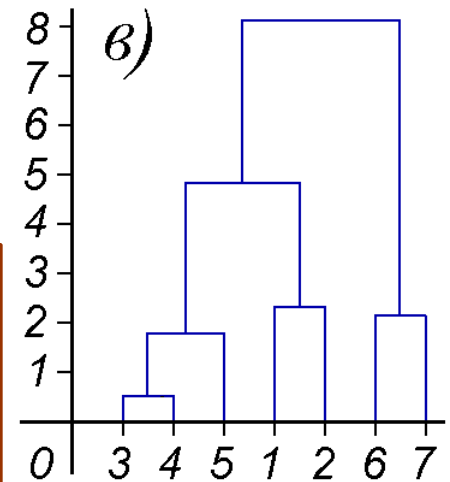
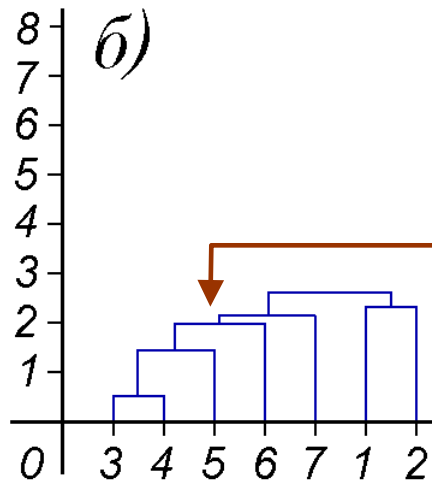
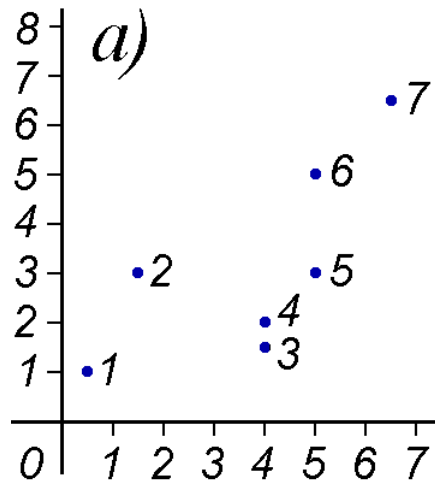
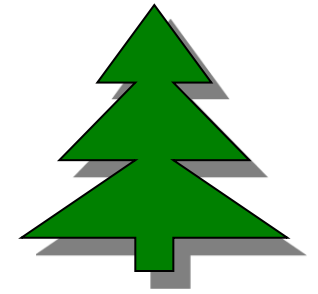
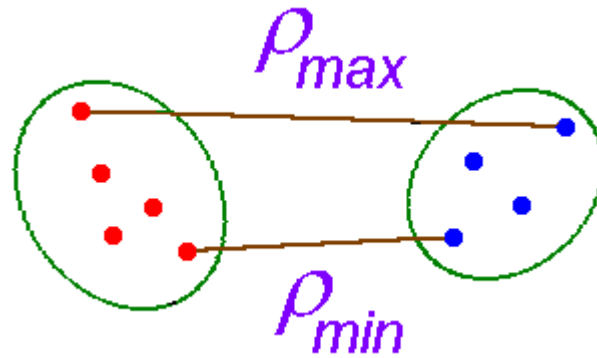
Недостатком этой процедуры является возможность появления *инверсий* — нарушений монотонности увеличения уровня при построении дендрограммы.

**P5)** МЕТОД УОРДА:  $\rho_W = \frac{n_k n_l}{n_k + n_l} |\bar{x}_k - \bar{x}_l|^2.$

Для этой процедуры инверсии невозможны.



# Процедуры «ближнего соседа» и «дальнего соседа». Дендрограммы.



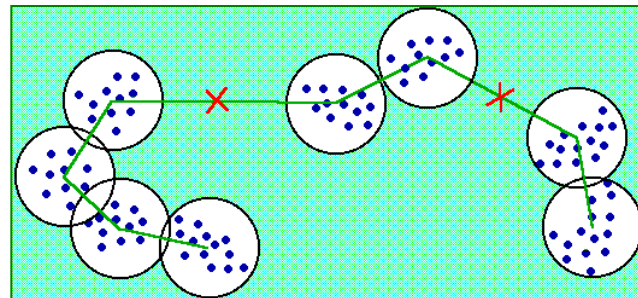
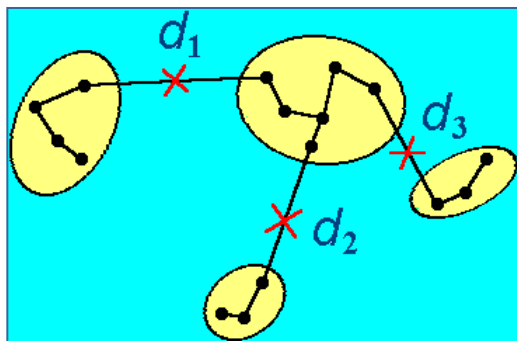
Метод «ближнего соседа» имеет недостаток — так называемый «цепочечный эффект»

# Практическое задание 4

- 1) Для строк таблицы `Simpl` командой `dist` постройте список попарных расстояний между объектами (он представляет собой записанные в вектор элементы из нижнетреугольной матрицы попарных расстояний), запишите его в переменную `d`
  - 2) Примените к `d` процедуру иерархической кластеризации `hclust` с аргументом `method="complete"` (обозначающим метод «дальнего соседа») и запишите результат в переменную `r`
  - 3) Постройте для `r` дендрограмму командой `plot`
  - 4) Визуально определите оптимальный уровень разрезания дендрограммы и распределите объекты по кластерам командой `cutree`. Запишите номера кластеров в вектор `clust`
  - 5) Постройте матричную диаграмму для таблицы `Simpl` с раскраской кластеров цветами из вектора `clust`
- 
- 6) Постройте график «осыпь», показывающий рост уровней, на которых происходили слияния ближайших кластеров (используйте для `r$height` команду `plot` с аргументом `type="l"`)



# Связные компоненты и алгоритм «Форель»



Для процедуры «ближнего соседа» разрезание дендрограммы на некотором уровне равносильно отбрасыванию наиболее длинных рёбер в графе связности объектов. В результате чего граф распадается на связные компоненты.

Предварительная очистка от «шумящих» объектов обычно позволяет существенно улучшить классификацию, получаемую методом «ближнего соседа».

Для большого числа объектов применяется также алгоритм «Форель», в котором близкие объекты предварительно объединяются в «шаровидные» группы.

# Кластеризация Big Data

Для таблиц, содержащих миллионы строк (объектов) и сотни столбцов (признаков), применим следующий подход:

- 1) Выбрать подмножество строк по некоторым условиям на признаки (например, в корпусе можно отбирать слова по частоте встречаемости, по части речи и т. п.)
- 2) Для каждого признака оценить по выборке (скажем, размера  $n = 1000$ ) нижнюю и верхнюю квартили и на их основе перекодировать значения признаков в  $-1, 0, 1$
- 3) Отобрать случайное подмножество строк (скажем, 10%) и для него составить список из, скажем, 1000 наиболее часто встречающихся кодировок строк. Кодировки можно экономно записать в битовой форме, что позволит оперировать с сотнями признаков. При подсчёте частот можно «склеивать» близкие кодировки
- 4) Для элементов из списка подсчитать частоты по всем объектам, отобрать наиболее часто встречающиеся, затем объединить их в кластеры с помощью, например, метода «ближнего соседа» с манхеттенской метрикой

# Главное в теме

- Перед кластеризацией строк таблицу данных, как правило, **нормируют** по столбцам (для очищенных от «выбросов» признаков применяют нормировку N1, для других — N3)
- Полезно визуально изучить данные с помощью построения **диаграмм рассеяния** всех пар признаков
- Перед кластеризацией иногда бывает полезно **удалить «шумящие» объекты**, т. е. находящиеся в «хвостовых» областях и в областях «разреженности» объектов
- Если число классов неизвестно, то обычно используют иерархические процедуры: методы **«дальнего соседа»** (complete) и метод **Уорда** (ward.D2) хорошо выделяют кластеры типа «ядра» или «сгущения в среднем», а метод **«ближнего соседа»** (single) — кластеры «типа ленты»
- Если число кластеров известно (нередко оно меньше 10), то применяют метод **К-средних**, который ориентирован на поиск кластеров типа «кластер с центром». Для таких кластеров важным показателем качества кластеризации служит большая доля **межкластерной инерции**
- Модификации метода К-средних (pam) и иерархических процедур (agnes), а также другие методы кластеризации реализованы в пакете **cluster**, который сопровождает книгу Kaufman L., Rousseeuw P. *"Finding Groups in Data"*



Научное исследование — это искусство,  
а правила в искусстве, если они слишком жёстки,  
приносят больше вреда, чем пользы.

*Дж. Томсон, «Дух науки»*



# Домашнее задание

- 1) Импортируйте файла CarSales.txt поместите столбцы Price и Engine\_s в таблицу данных d, вычислите число строк n в таблице d функцией nrow
- 2) Нормируйте столбцы таблицы d с помощью устойчивой нормировки N3
- 3) Напишите программу на языке R для выявления изолированных ("шумовых») объектов, которые имеют сумму расстояний до ближайших к ним k объектов больше, чем выборочная  $(1 - \alpha)$ -квантиль в векторе x, содержащем такие суммы расстояний для каждого объекта (строки d)  
(Используйте вложенные циклы, команду if, функцию sort для сортировки вектора t, содержащего k текущих ближайших квадратов расстояний, командой rep иницируйте компоненты вектора t символом Inf, обозначающим машинную бесконечность. *Идея алгоритма*: если квадрат расстояния до очередной точки меньше, чем  $t[k]$ , то максимум заменяется на этот квадрат расстояния и вектор сортируется.)
- 4) Положите  $k = 10$ ,  $\alpha = 0,1$  и запустите программу
- 5) Вычислите границу b, отделяющую "шумовые" объекты от остальных, затем командой plot постройте диаграмму рассеяния для таблицы d, раскрасив красным цветом (col=2) точки, соответствующие изолированным («шумовым») объектам
- 6) Измените значение параметра  $\alpha$  на 0,05 и повторите пункт 5

*Подсчитайте числа красных точек*