



Институт технологий управления

Программные средства анализа данных

01.03.05 Статистика

Профиль «Анализ данных в бизнесе и экономике»

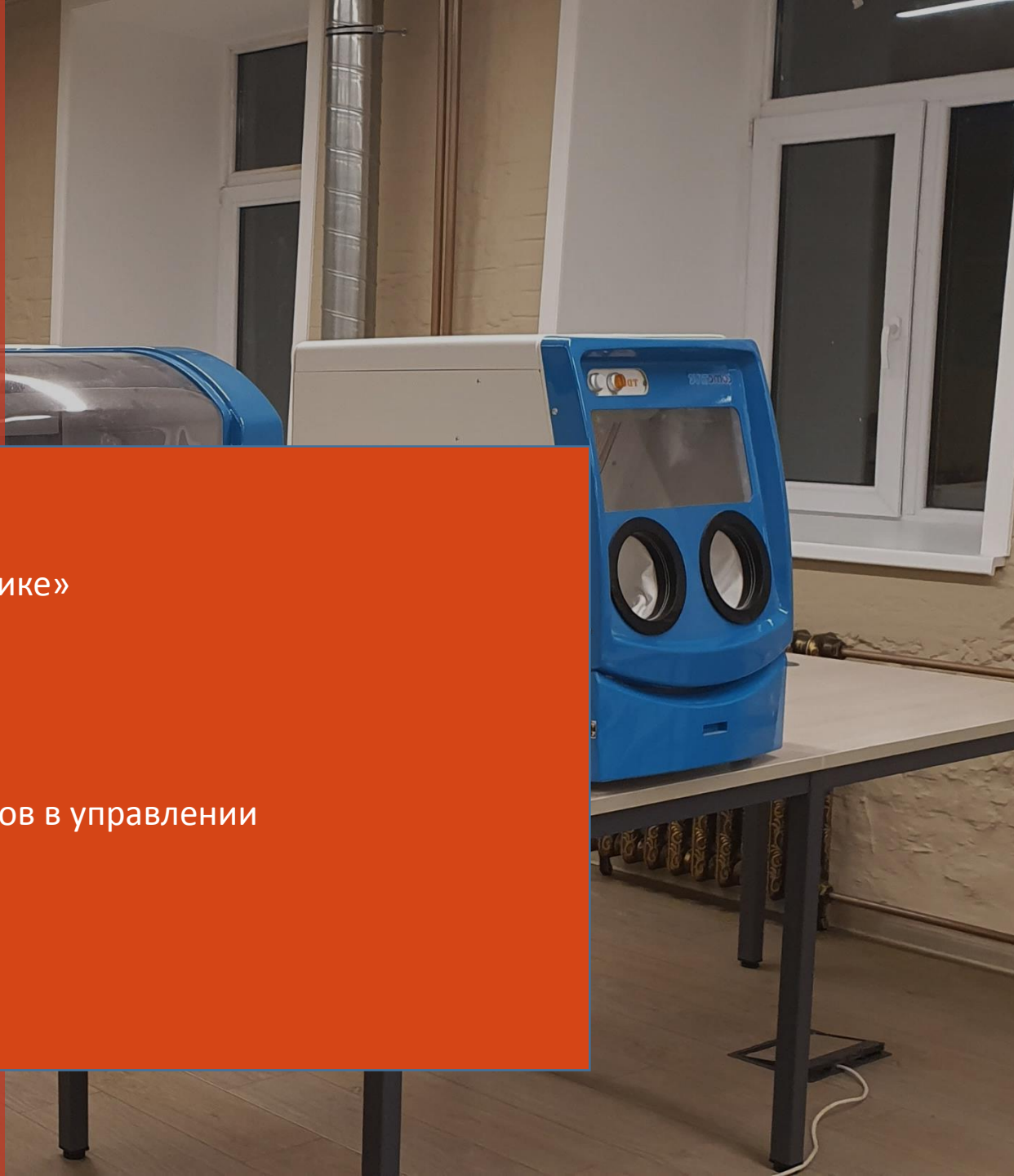
Квалификация «бакалавр»

Бурцева Татьяна Александровна

Профессор кафедры статистики и математических методов в управлении

Burceva\_t@mirea.ru

Москва, 2022



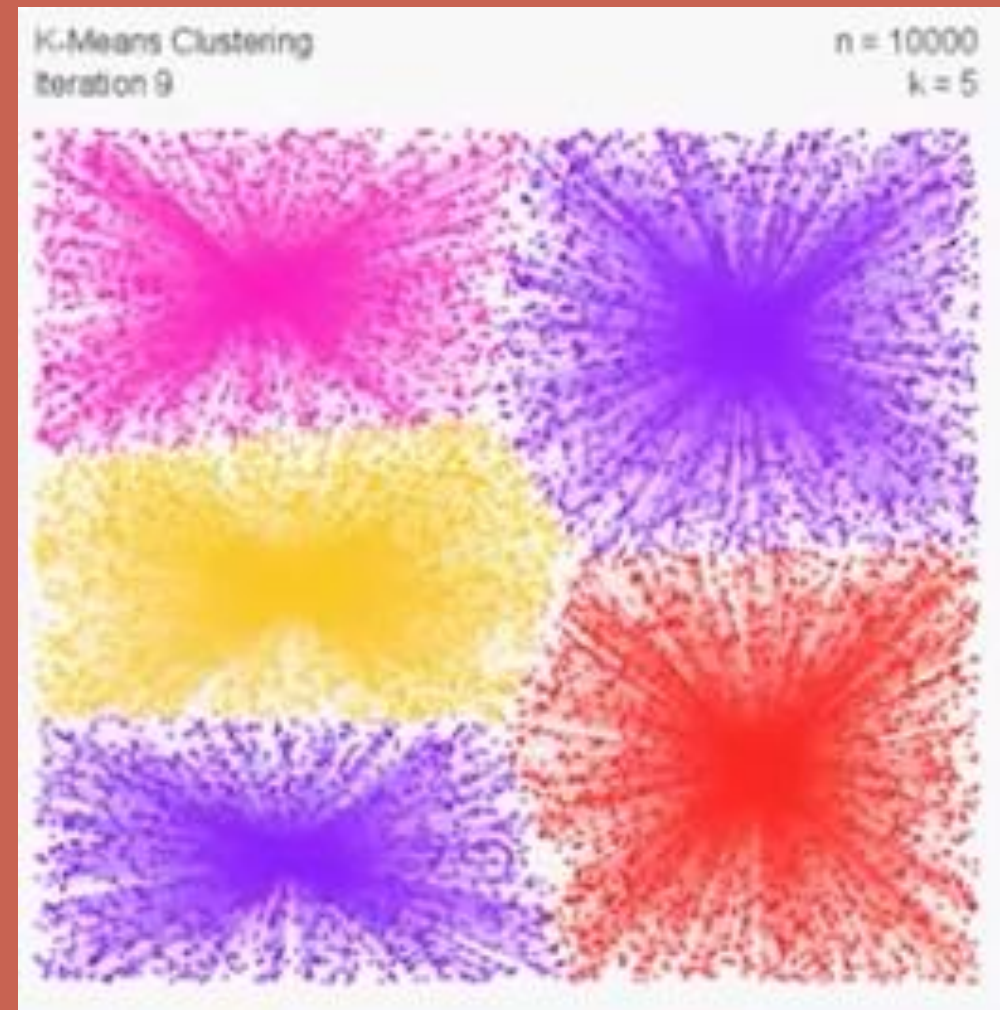
## Тема 8. Кластерный анализ в R

### План лекции

1. Понятие и назначение кластерного анализа.
2. Виды кластерного анализа в R.
3. Метод k-средних.
4. Методы иерархического кластерного анализа.
5. Метрики и команды в R.

## 1. Понятие и назначение кластерного анализа

**Понятие: кластерный анализ** - это совокупность методов классификации многомерных наблюдений или объектов, основанных на определении понятия расстояния между объектами с последующим выделением из них групп наблюдений (кластеров).



## 1. Понятие и назначение кластерного анализа

Назначение: решение задачи кластеризации

- Когда нужно преобразовать «горы» информации в пригодные для дальнейшего изучения группы, используют кластерный анализ.

Кластеризация (или кластерный анализ) — это задача разбиения множества объектов на группы, называемые кластерами. Внутри каждой группы должны оказаться «похожие» объекты, а объекты разных группы должны быть как можно более отличны. Главное отличие кластеризации от классификации состоит в том, что перечень групп четко не задан и определяется в процессе работы алгоритма.

# Преимущества метода

- позволяет разбивать многомерный ряд сразу по целому набору параметров;
- можно рассматривать данные практически любой природы (нет ограничений на вид исследуемых объектов);
- можно обрабатывать значительные объемы информации, резко сжимать их, делать компактными и наглядными;
- может применяться циклически (проводится до тех пор, пока не будет достигнут нужный результат; а после каждого цикла возможно значительное изменение направленности дальнейшего исследования).

# Недостатки метода

- состав и количество кластеров зависит от заданного критерия разбиения;
- при преобразовании исходного набора данных в компактные группы исходная информация может искажаться, отдельные объекты могут терять свою индивидуальность;
- часто игнорируется отсутствие в анализируемой совокупности некоторых значений кластеров.

# Характеристики кластера

*Кластер имеет следующие математические характеристики: центр, радиус, среднеквадратическое отклонение, размер кластера.*

*Центр кластера* - это среднее геометрическое место точек в пространстве переменных.

*Радиус кластера* - максимальное расстояние точек от центра кластера.

Кластеры могут быть перекрывающимися. В этом случае невозможно при помощи математических процедур однозначно отнести объект к одному из двух кластеров. Такие объекты называют спорными.

*Спорный объект* - это объект, который по мере сходства может быть отнесен к нескольким кластерам.

*Размер кластера* может быть определен либо по радиусу кластера, либо по среднеквадратичному отклонению объектов для этого кластера. **Объект относится к кластеру, если расстояние от объекта до центра кластера меньше радиуса кластера.** Если это условие выполняется для двух и более кластеров, объект является спорным.



# Среднеквадратичная ошибка разбиения

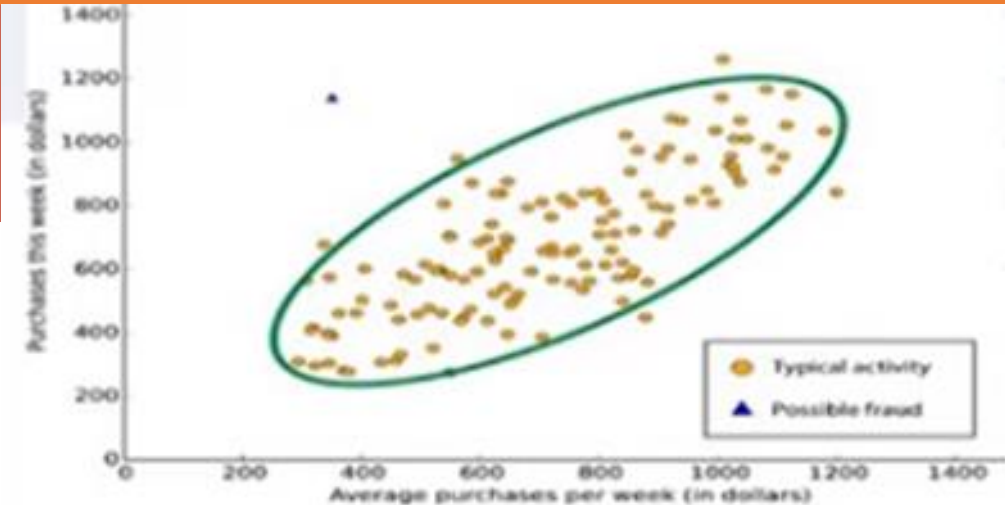
Задачу кластеризации можно рассматривать как построение оптимального разбиения объектов на группы. При этом оптимальность может быть определена как требование минимизации среднеквадратической ошибки разбиения:

$$e^2(X, L) = \sum_{j=1}^K \sum_{i=1}^{n_j} \|x_i^{(j)} - c_j\|^2$$

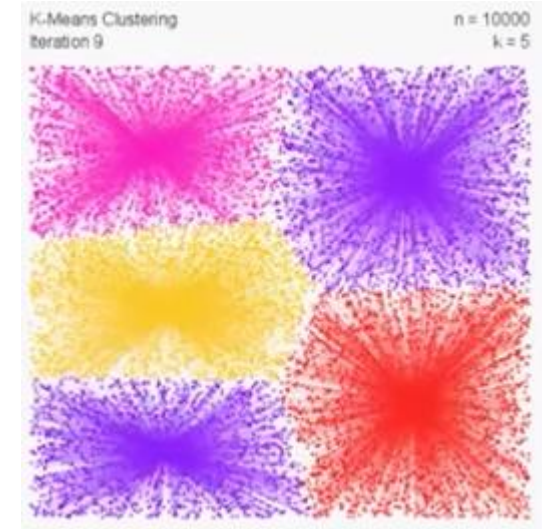
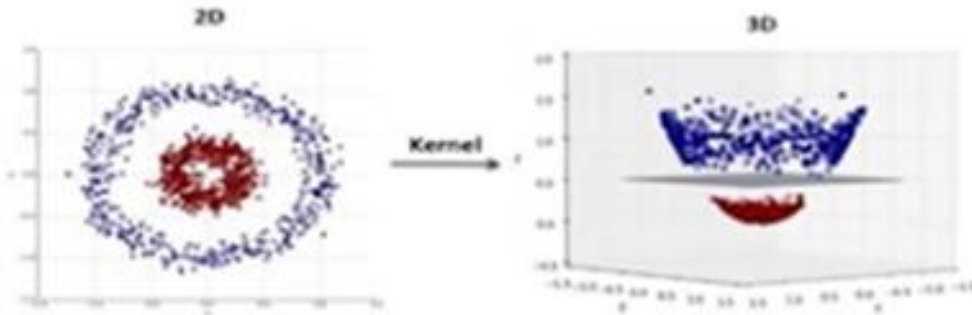
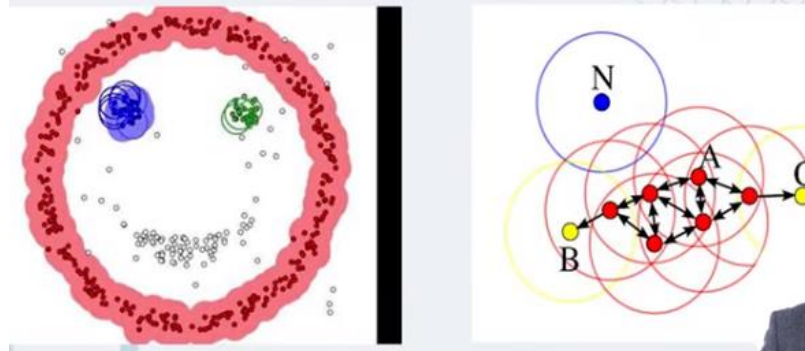
где  $c_j$  — «центр масс» кластера  $j$  (точка со средними значениями характеристик для данного кластера).



# Примеры кластеров, получаемых разными алгоритмами



DBSCAN (Density-based spatial clustering of applications with noise.)



Близость единичного элемента ~ DBSCAN

Максимальное расстояние до элемента или попарные расстояния

Минимизация энергии (например, суммарной дисперсии)

Среднее расстояние внутри кластера или дисперсия ~ kMeans

# Этапы кластерного анализа

Применение кластерного анализа в общем виде сводится к следующим этапам:

1. Отбор выборки объектов для кластеризации.
2. Определение множества переменных, по которым будут оцениваться объекты в выборке. При необходимости – нормализация значений переменных.
3. Вычисление значений меры сходства между объектами.
4. Применение метода кластерного анализа для создания групп сходных объектов (кластеров).
5. Представление результатов анализа.

После получения и анализа результатов возможна корректировка выбранной метрики и метода кластеризации до получения оптимального результата.

Обычной формой представления исходных данных в задачах кластерного анализа служит матрица:

$$X = \begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & x_{n3} & x_{n4} \end{pmatrix}$$

каждая строка которой, представляет результат измерений  $k$ , рассматриваемых признаков на одном из обследованных объектов.

- Из-за **неоднородности единиц измерения признаков** невозможно корректно рассчитать **расстояния** между точками. Эта проблема решается при помощи предварительной *стандартизации* переменных.
- **Стандартизация** (*standardization*) или нормирование (*normalization*) приводит значения всех преобразованных переменных к единому диапазону значений путем выражения через *отношение* этих значений к некой величине, отражающей определенные свойства конкретного признака.

# Нормализация/Стандартизация

- Нужна чтобы все компоненты давали одинаковый вклад при расчете «расстояния».
- В процессе нормализации все значения приводятся к некоторому диапазону, например,  $[-1, -1]$  или  $[0, 1]$ .

# Приёмы нормирования показателей

- $x_i^H = \frac{x_i}{\bar{x}}$
- Недостаток: значительные различия между объектами по одному из показателей могут существенным образом повлиять на значение интегрального индикатора, что допустимо, только если такой показатель имеет ключевое значение

## Приёмы нормирования показателей

- Метод максимум-минимум

- $x_i^H = \frac{x_i - x_{min}}{x_{max} - x_{min}}$  или  $x_i^H = 1 - \frac{x_i - x_{min}}{x_{max} - x_{min}}$

Две формулы необходимы для трансформации показателей разной направленности

Недостаток: позволяет исключить чрезмерное влияние одного частного показателя на интегральный, но при этом не позволяет учитывать серьёзные различия между объектами исследования в тех случаях, когда эти различия значимы



## Приёмы нормирования показателей

- Стандартизация позволяет сократить разброс между значениями показателя возможно путём логарифмирования значений показателей:

$$x_i^H = \frac{\log x_i - \log x_{\min}}{\log x_{\max} - \log x_{\min}}$$

Есть возможность адекватно учитывать различия между показателями по разбросу максимальных и минимальных значений, но высокая степень субъективности

# Приёмы нормирования показателей

Для показателей, имеющих отрицательную направленность, стандартизация показателей проводится по их обратным значениям по формуле:

$$x_i^H = \frac{\frac{1}{x_i}}{\max \frac{1}{x_i}}$$

Нормирование по схеме, приведённой ниже, обеспечивает сравнение не с лучшими объектами рейтингования, а с аутсайдерами:

$$x_i^H = 1 - \frac{x_i}{x_{\max}}$$

- Наряду со *стандартизацией*, существует вариант придания каждому наблюдению определённого веса, который бы отражал его *значимость*.
- В качестве весов могут выступать *экспертные оценки*, полученные в ходе опроса экспертов - *специалистов предметной области*.
- Полученные **произведения нормированных наблюдений на соответствующие веса** позволяют получать расстояния между точками в многомерном пространстве с учетом неодинакового их веса.

## 2. Виды кластерного анализа в R

# Алгоритмы кластеризации



Выбор конкретного метода кластерного анализа зависит от цели классификации.

# Методы кластерного анализа

- **Иерархические:** последовательно объединяют меньшие кластеры в большие или разделяют большие кластеры на меньшие.

Иерархические методы кластерного анализа используются при **небольших объемах наборов данных**.

**Преимуществом иерархических методов** кластеризации является их наглядность.

- **Неиерархические:** строят одно разбиение объектов на кластеры

Используя различные методы кластерного анализа, **аналитик** может получить различные решения для одних и тех же данных. Это считается нормальным явлением.

### 3. Метод k-средних

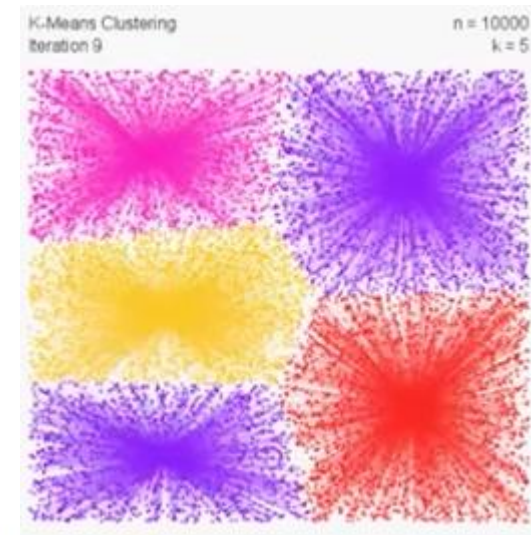
Итеративный алгоритм, основанный на минимизации суммарного квадратичного отклонения точек кластеров от центров этих кластеров.

предложен в 50-х годах XX в. Гуго Штейнгаузом (и почти одновременно Стюартом Ллойдом ) и получивший широкое распространение благодаря работе Маккуина, до сих пор остаётся одним из самых популярных алгоритмов кластеризации

Метод	Параметры	Результат	Достоинства	Недостатки
<i>k</i> -средних	Число кластеров ( $K$ )	Центры кластеров, объекты, размеченные номерами кластеров	Простота реализации, быстродействие (при небольших размерностях)	Чувствительность к «выбросам», искажающим среднее значение расстояния; чувствительность к размерности (количеству объектов); наличие входного параметра

# Алгоритмы квадратичной ошибки

К недостаткам данного алгоритма можно отнести необходимость задавать количество кластеров для разбиения.



Алгоритмы квадратичной ошибки относятся к типу плоских алгоритмов. Самым распространенным алгоритмом этой категории является метод k-средних. Этот алгоритм строит заданное число кластеров, расположенных как можно дальше друг от друга.



# АЛГОРИТМ МЕТОДА k-средних

Алгоритм разделительной кластеризации, основан на разбиении множества элементов векторного пространства на заранее определенное число кластеров  $k$ . Метод относится к неиерархическим алгоритмам кластеризации. Алгоритм представляет собой итерационную процедуру:

1. Выбирается число кластеров  $k$ .
2. Из исходного множества данных случайным образом выбираются  $k$  записей, которые будут служить начальными центрами кластеров.
3. Для каждой записи исходной выборки определяется ближайший к ней центр кластера. При этом записи, «притянутые» определенным центром, образуют начальные кластеры.
4. Вычисляются центроиды – центры тяжести кластеров. Каждый центроид – это вектор, элементы которого представляют собой средние значения признаков, вычисленные по всем записям кластера. Затем центр кластера смещается в его центроид.

Процесс итерации прекращается, когда границы кластеров не перестанут изменяться от итерации к итерации, т.е. на каждой итерации в каждом кластере будет оставаться один и тот же набор записей.

# library(cluster) – библиотека в R

kmeans(данные, число кластеров, число итераций) – команда в R для реализации метода k-средних («K-Means Clustering»)

## Проверка качества кластеризации

После получения результатов кластерного анализа методом k-средних следует проверить правильность кластеризации (т.е. оценить, насколько кластеры отличаются друг от друга). Для этого рассчитываются средние значения для каждого кластера. При хорошей кластеризации должны быть получены сильно отличающиеся средние для всех измерений или хотя бы большей их части.

Достоинства *алгоритма k-средних*:

- простота использования;
- быстрота использования;
- понятность и прозрачность алгоритма.

Недостатки *алгоритма k-средних*:

- алгоритм слишком чувствителен к выбросам, которые могут исказить среднее. Возможным решением этой проблемы является использование модификации алгоритма - алгоритм k-медианы;
- алгоритм может медленно работать на больших базах данных. Возможным решением данной проблемы является использование выборки данных.

# Пример в R применения метода k-средних

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Кластерный анализ регионов 2019г в ... t t1 tmod yt1 yt12 Model2.R x2 t2 f1

Filter

№	Регионы	x1	x2	x3	x4	x5	x6	x7	x8	x9	Y
1	Белгородская область	1.1662636	0.9783068	1.0087610	1.0362342	1.028	0.9217687	1.0084368	1.0368448	1.0841407	1.0121021
2	Брянская область	1.0188169	0.9908783	1.0023256	1.1931545	1.036	0.8612565	1.0254947	1.0174817	1.0795432	1.0593567
3	Владимирская область	1.1392618	0.9738903	0.9821429	1.1211304	1.022	1.0248447	0.9877089	1.1495744	0.8908879	1.0292936
4	Воронежская область	1.0471565	0.9960122	0.9942922	1.1173009	1.026	0.9932203	0.9910779	1.1134040	0.9476295	1.0355821
5	Ивановская область	1.2579042	1.0316898	1.0088384	1.0369892	1.029	0.9234828	1.0025269	0.9046176	0.8599605	1.0055646
6	Калужская область	1.1072581	1.0112878	0.9950125	1.0620433	1.027	1.0000000	1.0044239	1.0147272	0.7858876	1.0163236
7	Костромская область	1.1085708	1.0040744	0.9975758	1.0904305	1.037	0.8434579	0.9754629	1.0850778	0.9696536	1.0345070
8	Курская область	1.1306988	1.0371018	0.9987358	1.0251669	1.032	1.1029412	1.0212669	0.9361236	1.1660773	1.0121722
9	Липецкая область	1.1232877	0.9832979	0.9963855	0.9665499	1.029	1.0064935	0.9621477	1.3952723	0.9629408	0.9753457
10	Московская область	1.0193452	0.9800428	1.0228717	1.1181189	1.026	0.9847095	0.9783258	0.9595980	0.9097614	1.0449613
11	Орловская область	1.0940820	0.9861595	1.0061576	1.0779332	1.034	1.1198630	1.0025491	1.0541818	0.8988810	1.0409587
12	Рязанская область	1.0338732	1.0123354	1.0167742	1.0670618	1.029	1.1119134	0.9756559	1.0335240	0.9911712	1.0204988
13	Смоленская область	0.9149649	0.9689570	1.0087829	1.0590618	1.022	0.9508671	0.9797996	1.1962061	0.9576512	1.0286566
14	Тамбовская область	1.0842300	0.9884189	1.0147965	1.0335650	1.040	1.0060790	1.0001264	1.0983600	1.7135701	1.0239492
15	Тверская область	0.7811893	0.9507783	0.9974874	1.0415858	1.021	1.0406091	1.0023817	1.5202275	1.0346967	1.0222295

Showing 1 to 18 of 87 entries, 12 total columns

Environment History Connections

Global Environment

spanlist num [1:4] 0.15 0.5 0.8 1

t.delta 357.8

t.predict num [1:20] 765 771 773 7...

types chr [1:12] "text" "text"...

Files Plots Packages Help Viewer

Zoom Export

Console Terminal Jobs

```
> library(readxl)
> types = c("text", "text", rep("numeric", 10))
> t <- as.data.frame(read_excel("C:/Users/компьютер/Documents/Данные.xlsx", 1,
+                               col_types = types))
> view(t)
> |
```

Исходные данные в переменной t

# Скрипт в R – подготовка данных о регионах

```
# создание таблицы в переменной t
```

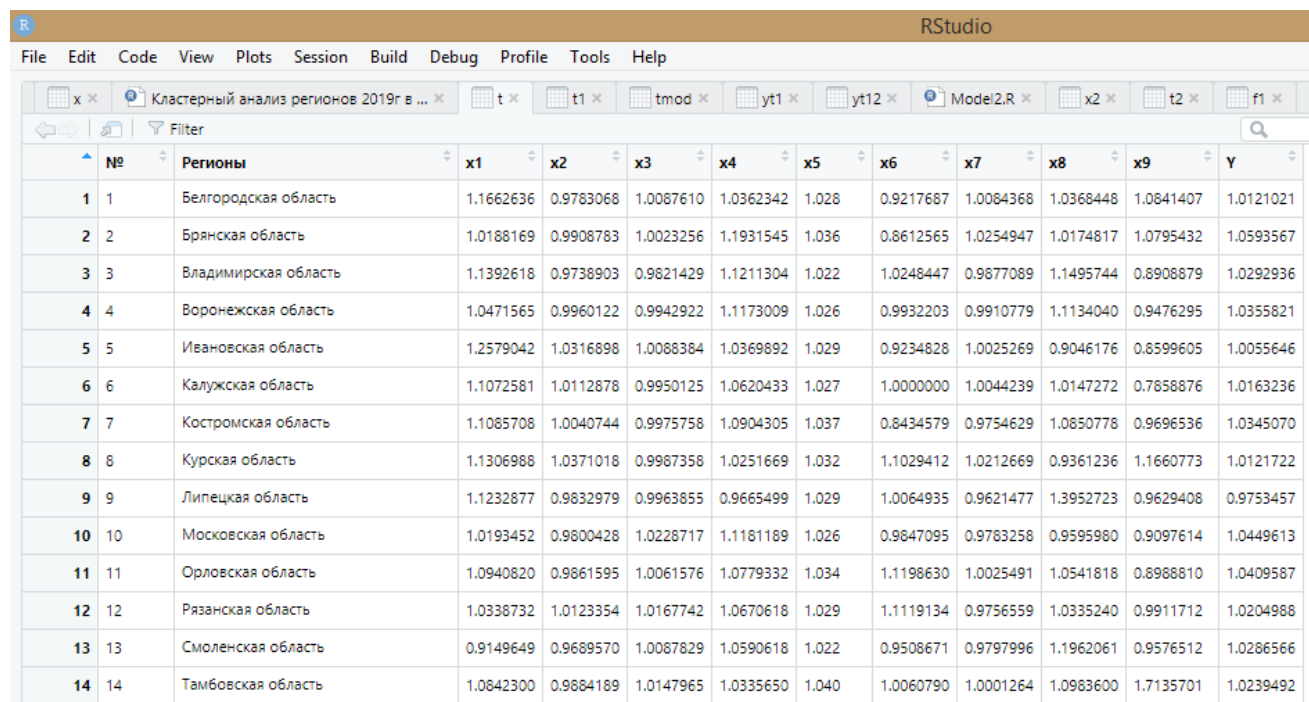
```
library(readxl)
```

```
types = c( "text", "text",  
rep("numeric", 10))
```

```
t <-
```

```
as.data.frame(read_excel("C:/Users/к  
омпьютер/Documents/Данные.xlsx",  
1, col_types = types))
```

```
View(t)
```



№	Регионы	x1	x2	x3	x4	x5	x6	x7	x8	x9	Y
1	Белгородская область	1.1662636	0.9783068	1.0087610	1.0362342	1.028	0.9217687	1.0084368	1.0368448	1.0841407	1.0121021
2	Брянская область	1.0188169	0.9908783	1.0023256	1.1931545	1.036	0.8612565	1.0254947	1.0174817	1.0795432	1.0593567
3	Владимирская область	1.1392618	0.9738903	0.9821429	1.1211304	1.022	1.0248447	0.9877089	1.1495744	0.8908879	1.0292936
4	Воронежская область	1.0471565	0.9960122	0.9942922	1.1173009	1.026	0.9932203	0.9910779	1.1134040	0.9476295	1.0355821
5	Ивановская область	1.2579042	1.0316898	1.0088384	1.0369892	1.029	0.9234828	1.0025269	0.9046176	0.8599605	1.0055646
6	Калужская область	1.1072581	1.0112878	0.9950125	1.0620433	1.027	1.0000000	1.0044239	1.0147272	0.7858876	1.0163236
7	Костромская область	1.1085708	1.0040744	0.9975758	1.0904305	1.037	0.8434579	0.9754629	1.0850778	0.9696536	1.0345070
8	Курская область	1.1306988	1.0371018	0.9987358	1.0251669	1.032	1.1029412	1.0212669	0.9361236	1.1660773	1.0121722
9	Липецкая область	1.1232877	0.9832979	0.9963855	0.9665499	1.029	1.0064935	0.9621477	1.3952723	0.9629408	0.9753457
10	Московская область	1.0193452	0.9800428	1.0228717	1.1181189	1.026	0.9847095	0.9783258	0.9595980	0.9097614	1.0449613
11	Орловская область	1.0940820	0.9861595	1.0061576	1.0779332	1.034	1.1198630	1.0025491	1.0541818	0.8988810	1.0409587
12	Рязанская область	1.0338732	1.0123354	1.0167742	1.0670618	1.029	1.1119134	0.9756559	1.0335240	0.9911712	1.0204988
13	Смоленская область	0.9149649	0.9689570	1.0087829	1.0590618	1.022	0.9508671	0.9797996	1.1962061	0.9576512	1.0286566
14	Тамбовская область	1.0842300	0.9884189	1.0147965	1.0335650	1.040	1.0060790	1.0001264	1.0983600	1.7135701	1.0239492

# АЛГОРИТМ МЕТОДА k-средних в R

```
#формируем таблицу только из переменных x2, x3
x<-t[,c(4, 5)]
View(x)
#нормируем данные
for (i in 1:2){
  minx<-min(x[,i])
  maxx<-max(x[,i])
  x[,i]<-(x[,i]-minx)/(maxx-minx)
}
#называем строки именем региона
rownames(x) <- t$Регионы
View(x)
#загружаем библиотеку и вызываем команду, 2 кластера и 4 итерации,
library(cluster)
kmeans(x, 2, 4)
#визуализируем
km=kmeans(x, 2, 4)
plot(x, col=km$cluster)
```

	x2	x3
Белгородская область	0.30537212	0.4713269
Брянская область	0.34539809	0.4139119
Владимирская область	0.29131068	0.2338459
Воронежская область	0.36174364	0.3422401
Ивановская область	0.47533644	0.4720177
Калужская область	0.41037942	0.3486658
Костромская область	0.38741286	0.3715349
Курская область	0.49256765	0.3818844
Липецкая область	0.32126313	0.3609161
Московская область	0.31089919	0.5972196
Орловская область	0.33037409	0.4481006
Рязанская область	0.41371476	0.5428193
Смоленская область	0.27560342	0.4715230
Тамбовская область	0.33756756	0.5251752

# Результаты

```
within cluster sum of squares by cluster:  
[1] 0.5901269 2.7719377  
(between_SS / total_SS = 24.6 %)
```



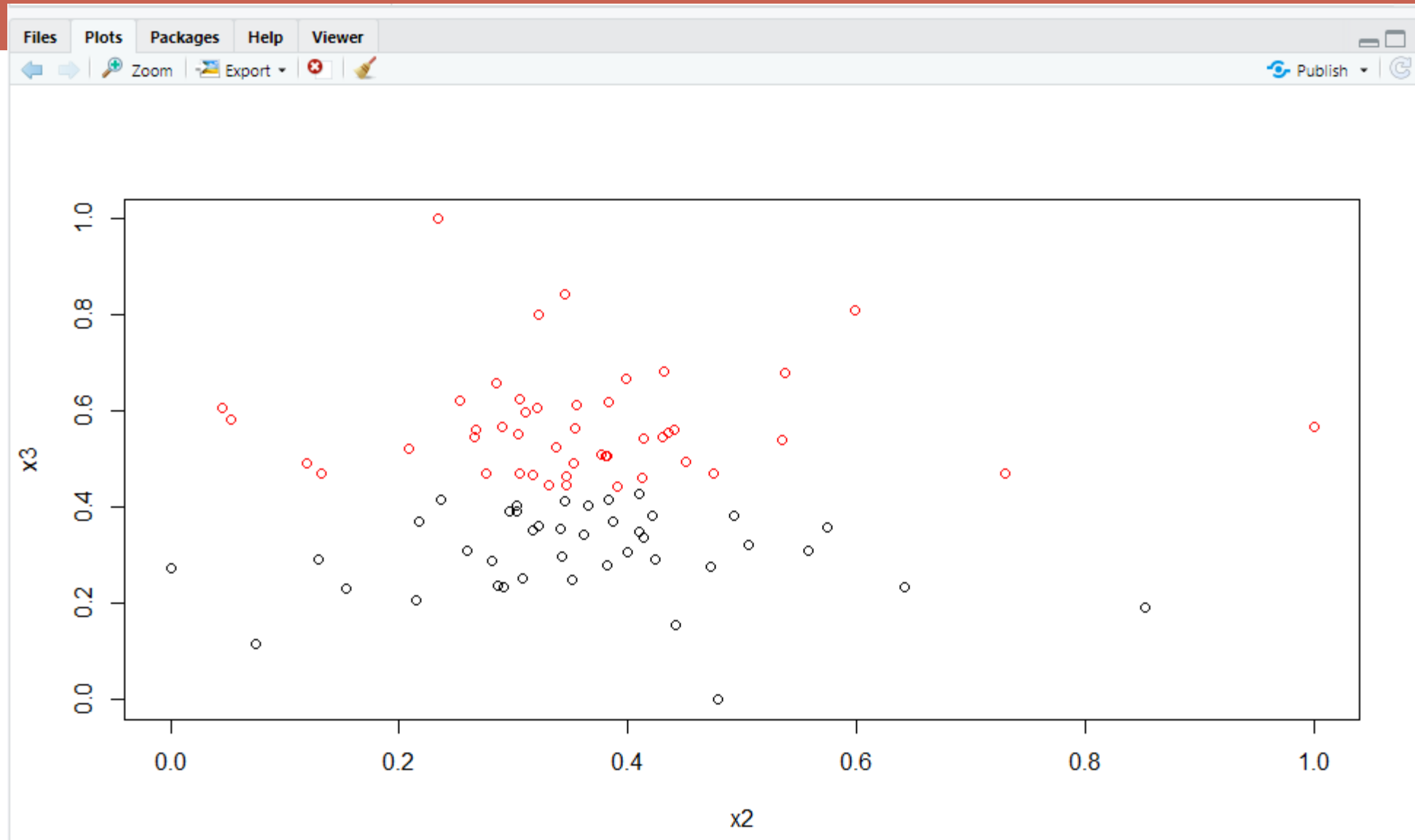
```
> library(cluster)  
> kmeans(x, 2, 4)  
K-means clustering with 2 clusters of sizes 12, 75
```

```
Cluster means:  
      x2      x3  
1 0.5977053 0.2970958  
2 0.3220267 0.4708044
```

```
Clustering vector:
```

Белгородская область	Брянская область	Владимирская область
2	2	2
Воронежская область	Ивановская область	Калужская область
2	2	2
Костромская область	Курская область	Липецкая область
2	1	2
Московская область	Орловская область	Рязанская область
2	2	2
Смоленская область	Тамбовская область	Тверская область
2	2	2
Тульская область	Ярославская область	г. Москва
2	2	1
Республика Карелия	Республика Коми	Архангельская область
2	2	2
Ненецкий авт. округ	Архангельская область без авт. округа	Вологодская область
2	2	2
Калининградская область	Ленинградская область	Мурманская область
2	2	2
Новгородская область	Псковская область	г. Санкт-Петербург
2	2	2
Республика Адыгея (Адыгея)	Республика Калмыкия	Республика Крым
2	1	2
Краснодарский край	Астраханская область	Волгоградская область
2	2	2
Ростовская область	г. Севастополь	Республика Дагестан
2	1	2
Республика Ингушетия	Кабардино-Балкарская Республика	Карачаево-Черкесская Республика
2	2	1

# Результаты





# Let cluster <- kmeans(x, k)

Список параметров переменной **cluster**

с результатами **kmeans** содержит элементы:

cluster \$ cluster: указывает кластер каждого наблюдения

cluster \$ centers : кластерные центры

cluster \$ totss: общая сумма квадратов отклонений от среднего

cluster \$ **tot.withinss**: сумма квадратов отклонений внутри кластера от его центра

Cluster\$betweenss: общая сумма квадратов минус предыдущий параметр

cluster \$ size: количество наблюдений в каждом кластере

## Value

`kmeans` returns an object of class "kmeans" which has a `print` and a `fitted` method. It is a list with at least the following components:

<code>cluster</code>	A vector of integers (from 1 : k) indicating the cluster to which each point is allocated.
<code>centers</code>	A matrix of cluster centres.
<code>totss</code>	The total sum of squares.
<code>withinss</code>	Vector of within-cluster sum of squares, one component per cluster.
<code>tot.withinss</code>	Total within-cluster sum of squares, i.e. <code>sum(withinss)</code> .
<code>betweenss</code>	The between-cluster sum of squares, i.e. <code>totss-tot.withinss</code> .
<code>size</code>	The number of points in each cluster.
<code>iter</code>	The number of (outer) iterations.
<code>ifault</code>	integer: indicator of a possible algorithm problem – for experts.

# Выбор оптимального числа кластеров

- #функция для выбора оптимального числа кластеров
- `kmean_withinss <- function(k) {`
- `cluster <- kmeans(x, k)`
- `return (cluster$tot.withinss)`
- `}`

```
# установим максимальное число кластеров
```

```
max_k <- 20
```

```
# выполним алгоритм от 2 до 20 кластеров
```

```
wss <- sapply(2:max_k, kmean_withinss)
```

```
elbow <- data.frame(2:max_k, wss)
```

```
# нарисуем график зависимости ошибки от числа кластеров
```

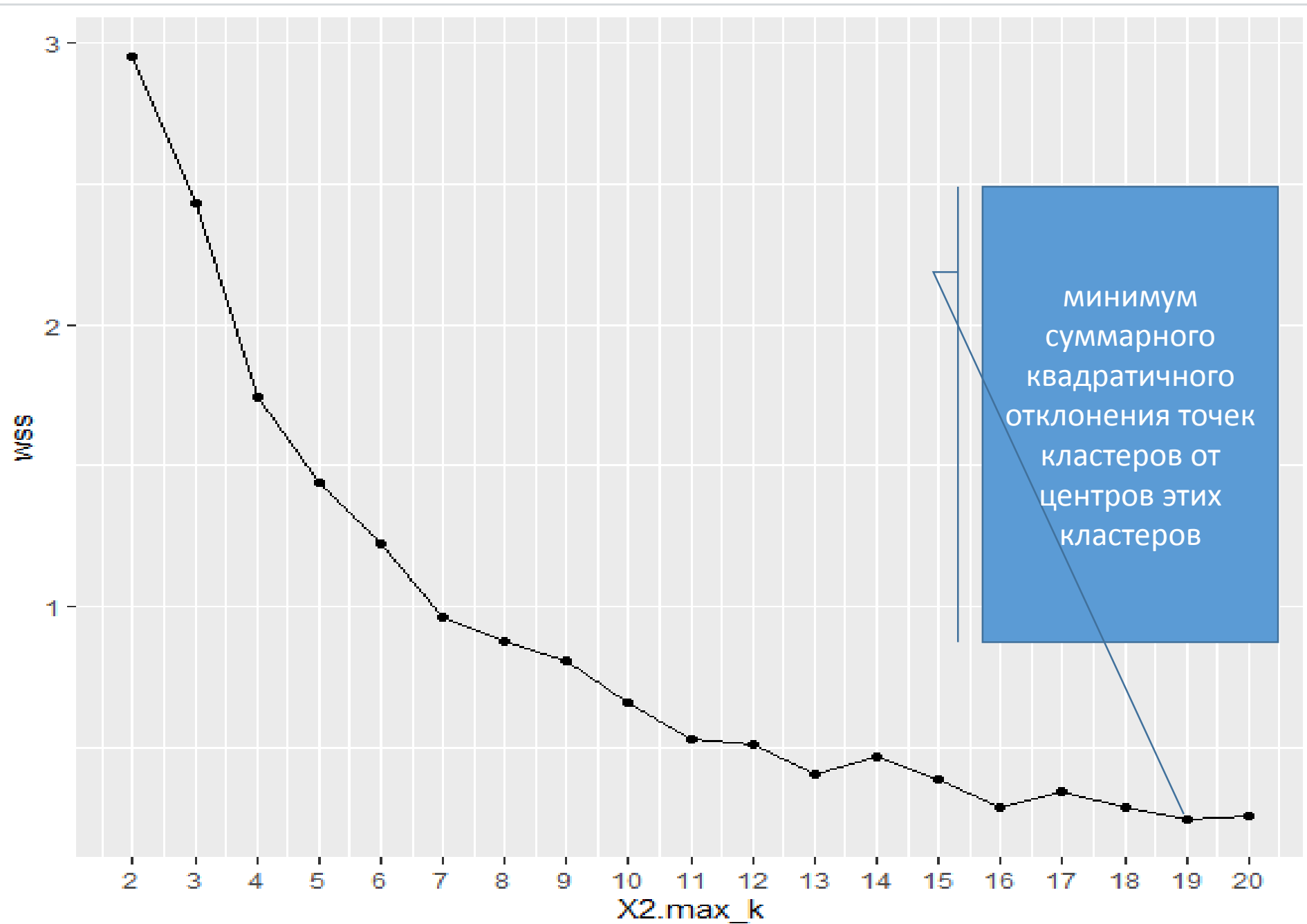
```
library(ggplot2)
```

```
ggplot(elbow, aes(x = X2.max_k, y = wss)) +
```

```
  geom_point() +
```

```
  geom_line() +
```

```
  scale_x_continuous(breaks = seq(1, 20, by = 1))
```



# для примера выбираем оптимальное число  
кластеров 18  
cluster = kmeans(x, 18)  
str(cluster)

```
k-means clustering with 18 clusters of sizes 1, 4, 5, 4, 3, 5, 5, 5, 6, 6, 3, 12, 2, 7, 5, 2, 6, 6
```

```
cluster means:
```

	x2	x3
1	0.85253975	0.19131312
2	0.08920724	0.22857199
3	0.11143614	0.53524366
4	0.56983121	0.30680745
5	0.52253051	0.72485215
6	0.42523772	0.36479218
7	0.33644174	0.34190817
8	0.40603923	0.28136909
9	0.32001907	0.46166459
10	0.36365576	0.51779779
11	0.29977388	0.88168852
12	0.31139755	0.60354491
13	0.86525528	0.51847863
14	0.45421017	0.53082963
15	0.27122850	0.39540504
16	0.46032757	0.07830048
17	0.27346304	0.25511545
18	0.38445129	0.42834425

```
within cluster sum of squares by cluster:
```

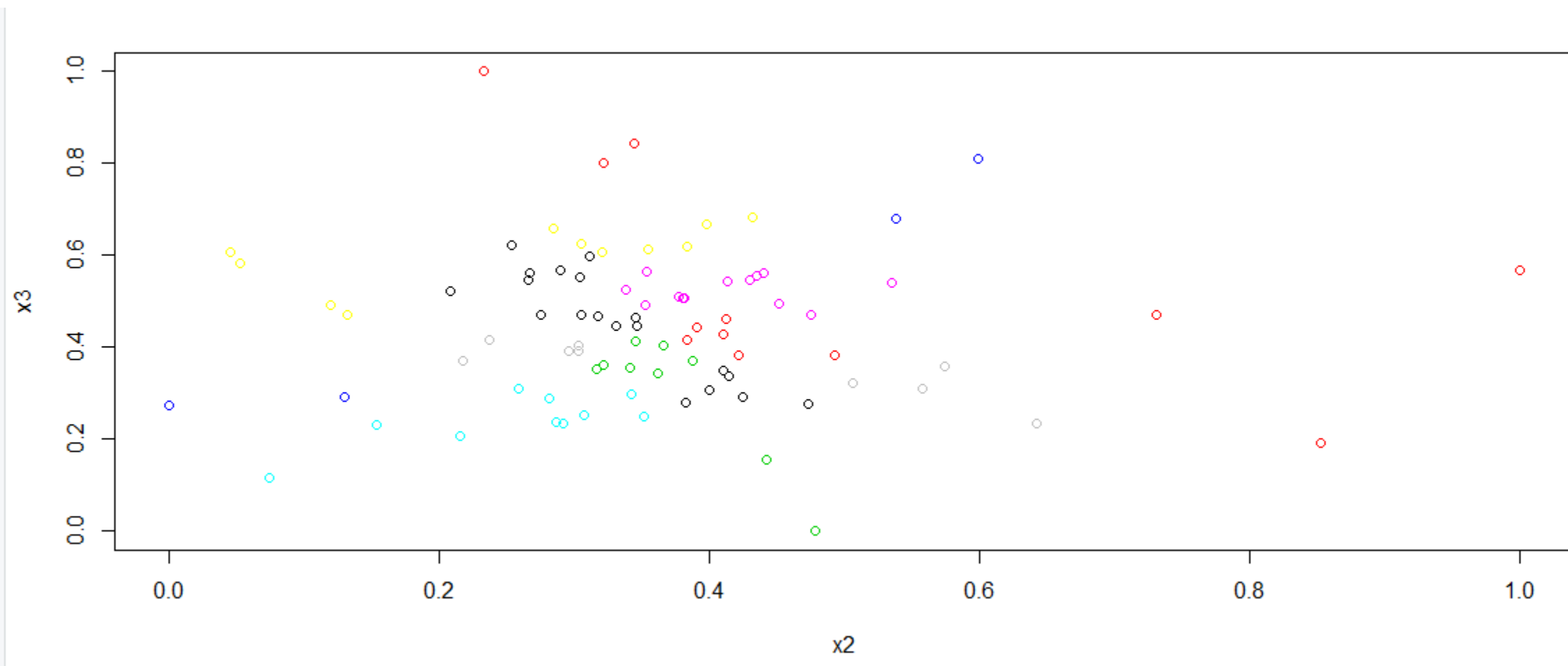
```
[1] 0.000000000 0.032978816 0.031322010 0.017547370 0.024986221 0.007906849 0.003994943 0.010174316 0.004290889  
[10] 0.004826396 0.028829975 0.040792743 0.041014054 0.016655804 0.007780643 0.012925018 0.012752006 0.005676943  
(between_SS / total_SS = 93.2 %)
```

List of 9

```
$ cluster      : int [1:87] 10 1 7 13 17 13 1 15 4 5 ...
$ centers      : num [1:18, 1:2] 0.389 0.148 0.46 0.292 0.271 ...
..- attr(*, "dimnames")=List of 2
.. ..$ : chr [1:18] "1" "2" "3" "4" ...
.. ..$ : chr [1:2] "x2" "x3"
$ totss       : num 4.46
$ withinss    : num [1:18] 0.0109 0.0176 0.0129 0.0166 0.0139 ...
$ tot.withinss: num 0.281
$ betweenss   : num 4.18
$ size        : int [1:18] 8 3 2 8 7 2 7 3 7 6 ...
$ iter        : int 4
$ ifault      : int 0
- attr(*, "class")= chr "kmeans"
```

> |

# Вывод: практически нет пересечений





#### 4. Методы иерархического кластерного анализа.

Под иерархической кластеризацией понимается подход, предполагающий построение дендрограммы (дерева разбиения множества объектов).

В отличие от метода  $k$ -средних, число кластеров при иерархической кластеризации не является постоянным, оно определяется в процессе работы алгоритма

Основой для работы иерархического метода кластеризации является так называемая матрица схожести. На каждом шаге алгоритма наиболее схожие объекты объединяются в один кластер (формируется ветвь дерева).

Иерархические методы различаются используемыми правилами группировки объектов. Наиболее известны следующие из них.:

**Метод ближнего соседа или одиночная связь.** Здесь *расстояние* между двумя кластерами определяется расстоянием между двумя наиболее близкими объектами (ближайшими соседями) в различных кластерах. Этот метод позволяет выделять кластеры сколь угодно сложной формы при условии, что различные части таких кластеров соединены цепочками близких друг к другу элементов. В результате работы этого метода кластеры представляются длинными "цепочками" или "волокнистыми" кластерами, "сцепленными вместе" только отдельными элементами, которые случайно оказались ближе остальных друг к другу.

**Метод наиболее удаленных соседей или полная связь.** Здесь расстояния между кластерами определяются наибольшим расстоянием между любыми двумя объектами в различных кластерах (т.е. "наиболее удаленными соседями"). Метод хорошо использовать, когда объекты действительно происходят из различных "рощ". Если же кластеры имеют в некотором роде удлинненную форму или их естественный тип является "цепочечным", то этот метод не следует использовать.

**Метод Варда (Ward's method).** В качестве расстояния между кластерами берется прирост суммы квадратов расстояний объектов *до центров кластеров*, получаемый в результате их объединения (Ward, 1963). В отличие от других методов кластерного анализа для оценки расстояний между кластерами, здесь используются методы *дисперсионного анализа*. На каждом шаге алгоритма объединяются такие два кластера, которые приводят к минимальному увеличению целевой функции, т.е. внутригрупповой суммы квадратов. Этот метод направлен на *объединение* близко расположенных кластеров и "стремится" создавать кластеры малого размера.

**Метод невзвешенного попарного среднего** (метод невзвешенного попарного арифметического среднего - *unweighted pair-group method using arithmetic averages*, UPGMA (Sneath, Sokal, 1973)).

В качестве расстояния между двумя кластерами берется среднее *расстояние* между всеми парами объектов в них. Этот метод следует использовать, если объекты действительно происходят из различных "рощ", в случаях присутствия кластеров "цепочного" типа, при предположении неравных размеров кластеров.

**Метод взвешенного попарного среднего** (метод взвешенного попарного арифметического среднего - *weighted pair-group method using arithmetic averages*, WPGMA (Sneath, Sokal, 1973)). Этот метод похож на метод невзвешенного попарного среднего, разница состоит лишь в том, что здесь в качестве весового коэффициента используется *размер кластера* (число объектов, содержащихся в кластере).

Этот метод рекомендуется использовать именно при наличии предположения о кластерах разных размеров.

**Невзвешенный центроидный метод** (метод невзвешенного попарного центроидного усреднения - *unweighted pair-group method using the centroid average* (Sneath and Sokal, 1973)).

В качестве расстояния между двумя кластерами в этом методе берется *расстояние* между их центрами тяжести.

**Взвешенный центроидный метод** (метод взвешенного попарного центроидного усреднения - *weighted pair-group method using the centroid average*, WPGMC (Sneath, Sokal 1973)). Этот метод похож на предыдущий, разница состоит в том, что для учета разницы между *размерами кластеров* (в числе объектов в них), используются веса. Этот метод предпочтительно использовать в случаях, если имеются предположения относительно существенных отличий в *размерах кластеров*.

**Алгоритм DBSCAN.** В предложенном в методе DBSCAN (DensityBased Spatial Clustering of Applications with Noise, то есть плотностной алгоритм пространственной кластеризации с присутствием шума) предполагается, что кластеры представляют собой некоторые плотные «сгустки» точек.

Как и в методах иерархической кластеризации, здесь используется матрица схожести. Кроме того, в методе DBSCAN используются следующие понятия:  $\epsilon$ -окрестность объекта  $x$   $U(x, \epsilon) = \{y \in V : \rho(x, y) \leq \epsilon\}$ , корневой (или ядерный) объект степени  $M$  (для заданного  $\epsilon$ ) – это такой объект,  $\epsilon$ -окрестность которого содержит не менее  $M$  других объектов. При заданном значении  $M$  говорят, что объект  $y$  непосредственно плотнодостижим из объекта  $x$ , если  $y \in U(x, \epsilon)$ , а объект  $x$  является корневым. Говорят, что объект  $y$  плотно-достижим из объекта  $x$ , если существуют такие объекты  $x_1, \dots, x_n$ , где  $x_1 = x$ ,  $x_n = y$ , что при всех  $i = 1, \dots, n - 1$  объект  $x_{i+1}$  непосредственно плотно-достижим из  $x_i$ . Заметим, что алгоритм DBSCAN определяет число кластеров  $K$  в процессе работы.

Опишем алгоритм DBSCAN.

Шаг 0. Зададим значения параметров  $\epsilon$  и  $M$ , положим  $K := 0$ .

Шаг 1. Если все объекты  $x \in V$  уже просмотрены, останов. В противном случае выбирается любой из них и отмечается как просмотренный.

Шаг 2. Если  $x$  – корневой объект, то создаём новый кластер (при этом полагаем  $K := K + 1$ ) и переходим к Шагу 3; в противном случае точка  $x$  помечается как «шум» (заметим, что впоследствии эта точка может оказаться в  $\epsilon$ -окрестности некоторой другой точки и быть включённой в один из кластеров) и переходим к Шагу 1.

Шаг 3. В созданный кластер включаются все объекты, которые являются плотно-достижимыми из (корневого) объекта  $x$ , после чего происходит переход к Шагу 1.

# Алгоритм MaxFlow

- Метод кластеризации, основанный на построении максимального потока (минимального разреза) в сети.
- Принцип, лежащий в основе алгоритма MaxFlow, состоит в том, чтобы строить разрезы связных компонент сети так, чтобы наиболее удалённые друг от друга вершины оказались бы в разных кластерах.
- Общая схема алгоритма MaxFlow выглядит следующим образом.  
Вход: Граф, для которого известны множество вершин  $V$  и множество рёбер  $E$ .
- Выход: Построенное разбиение множества  $V$  на кластеры.

*Шаг 0.* Зададим значение параметра  $\varepsilon > 0$  (см. неравенство (8)). Положим номер итерации метода  $h := 0$ . Положим  $V(h) := V$ ,  $E(h) := E$ , обозначим  $N(h) := \{V(h), E(h)\}$  – подграф, используемый на итерации  $h$ .

Для  $h = 0, 1, \dots$

*Шаг 1.* Найдём наиболее удалённые друг от друга вершины сети  $N(h)$ :

$$i^* \in V(h), j^* \in V(h) : (i^*, j^*) \in E(h); \quad d(i^*, j^*) = \max_{(i,j) \in E(h)} d(i, j). \quad (9)$$

Одну из найденных вершин будем считать источником, другую – стоком. Найдём минимальный разрез сети  $N(h)$ , удалим все входящие в него рёбра (получая тем самым некоторое разбиение  $C_h$ ).

*Шаг 2.* Проверим выполнение критерия останова (критерия качества кластеризации): если для кластеризации  $C_h$  выполнено неравенство (8), то метод останавливает свою работу (при этом кластеризация  $C_h$  рассматривается как решение задачи); в противном случае перейдём к Шагу 3.

*Шаг 3.* Выберем худшую (относительно используемого критерия  $q$ ) связную компоненту  $w_h$  сети  $N(h)$ , определим  $V(h+1)$  и  $E(h+1)$  как множества вершин

и рёбер компоненты  $w_h$ , положим  $N(h+1) := \{V(h+1), E(h+1)\}$  и перейдём к Шагу 1 при  $h := h+1$ .

Общая схема алгоритма MaxFlow описана.

При численной реализации алгоритма MaxFlow для каждого фиксированного разбиения вида (6) использовались следующие функции:

– среднее внутрикластерное расстояние:

$$r = \frac{1}{K} r_k, \quad \text{где} \quad r_k = \frac{1}{|V_k|} \sum_{(i,j) \in V_k} \rho(i,j), \quad k = 1, \dots, K, \quad (10)$$

$\rho(i,j)$  – расстояние (в выбранной метрике) между вершинами  $i$  и  $j$ ;

– среднее межкластерное расстояние:

$$R = \frac{1}{K} R_k, \quad \text{где} \quad R_k = \frac{1}{|V_k|} \sum_{i \in V_k} \rho(i, z_k); \quad z_k = \frac{1}{|V_k|} \sum_{i \in V_k} x_i; \quad k = 1, \dots, K, \quad (11)$$


---



$x_i$  – вектор координат вершины  $i$  (в пространстве факторов);  $z_k$  – центроид кластера  $k$ ,  $k = 1, \dots, K$ .

Функция  $Q$ , используемая в неравенствах (7) и (8), имела вид:

$$Q = r/R, \tag{12}$$

функция  $\rho$  представляет собой евклидово расстояние.

В качестве критерия качества компоненты связности  $w$  (функция  $q$ ) использовалась длина максимального ребра в этой компоненте; мы говорим, что компонента  $w$  хуже, чем компонента  $v$ , если  $q(w) > q(v)$ .

Значение параметра  $\varepsilon$  полагалось равным 0.05. Заметим, что в алгоритме MaxFlow количество кластеров  $K$  не задаётся заранее – оно определяется в процессе работы алгоритма и зависит от значения  $\varepsilon$  (так, при  $\varepsilon = 0$  получаем тривиальное решение задачи кластеризации, где каждый кластер содержит один элемент).



Метод	Параметры	Результат	Достоинства	Недостатки
Ward	Матрица схожести (для её вычисления используется функция )	Дендрограмма (уровень иерархии можно регулировать)	Оптимальность разбиения (по критерию минимизации прироста суммы квадратов внутрикластерных расстояний); удобство представления результата (дендрограмма)	Сложность реализации; высокая вычислительная сложность, большие затраты времени
DBSCAN	Матрица схожести, числа $\epsilon$ , $M$	Объекты, размеченные номерами кластеров или «нулевой» меткой («шум»)	Не требуется задавать количество кластеров; нечувствительность к форме кластеров; способность обнаружения «шумов»	Чувствительность к выбору метрики; чувствительность к дисперсии внутрикластерных расстояний

Метод	Параметры	Результат	Достоинства	Недостатки
MaxFlow	Нет	Набор списков номеров объектов, отнесённых к соответствующим кластерам	Отсутствие входных параметров; способность обнаружения «шумов»; нечувствительность к размерам и форме кластеров	Требуется предварительная сортировка исходного набора данных

Как показывают полученные результаты, алгоритм  $k$ -средних преимущественно «распознаёт» кластеры сферической формы, в то время как методы DBSCAN и MaxFlow менее чувствительны к форме кластеров.

# Выводы:

- 1. Если форма кластеров заранее неизвестна, то целесообразно использовать алгоритмы MaxFlow и DBSCAN;
- 2. Для сферических кластеров целесообразно применять метод k - средних.
- 3. Если необходимо минимизировать время решения задачи, а набор данных не слишком велик (несколько сотен объектов), то следует использовать алгоритм k -средних.
- 4. В случае иерархической кластеризации при наличии ограничений на уровень разбиения данных следует использовать алгоритм Варда.
- 5. При необходимости выявления «шумовых выбросов» следует применять алгоритм DBSCAN или MaxFlow.

# Иерархический кластерный анализ на R

Функции, которые используют в R для проведения иерархической кластеризации:

```
hclust(d, method = "complete")  
agnes(x, metric = "euclidean", stand = FALSE, method = "average")  
diana(x, metric = "euclidean", stand = FALSE)
```

где: d: матрица дистанций, полученная функцией `dist()` или как-то иначе;

method: метод кластеризации, определяемый одним из значений "ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median" или "centroid";

x: таблица данных для вычисления расстояний (наблюдения - по строкам, признаки - по столбцам);

metric: метрика расстояний между наблюдениями, т.е. "euclidean" (Евклидово) или "manhattan" (Манхеттен);

stand = TRUE: осуществляется стандартизация каждой переменной таблицы (по столбцам).

## Прошлый пример:

- #формируем матрицу расстояний

```
m <- dist(scale(x))
```

```
#применим иерархический метод Варда, чтобы не задавать число кластеров, а найти его
```

```
hc <- hclust(m^2, method = "ward.D")
```

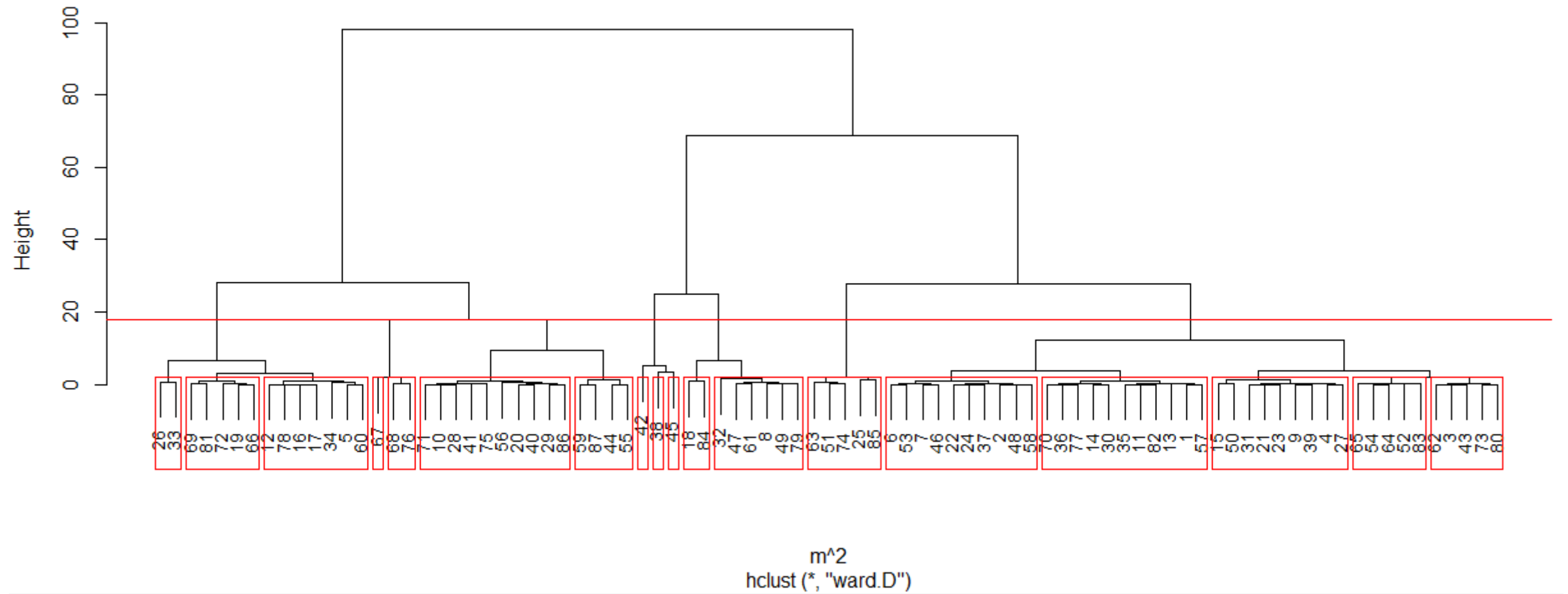
```
#построим дендограмму
```

```
plot(hc, cex = 0.9)
```

```
abline(h = 18, col = "red") # h - horizontal line, col - color
```

```
rect.hclust(hc, k = 18)
```

# Cluster Dendrogram



## 5. Метрики и команды в R

- Выбор метрики полностью лежит на исследователе, поскольку результаты кластеризации могут существенно отличаться при использовании разных мер.

### 1. Евклидово расстояние

Наиболее распространенная функция расстояния. Представляет собой геометрическим расстоянием в многомерном пространстве:

$$\rho(x, x') = \sqrt{\sum_i^n (x_i - x'_i)^2}$$

### 2. Квадрат евклидова расстояния

Применяется для придания большего веса более отдаленным друг от друга объектам. Это расстояние вычисляется следующим образом:

$$\rho(x, x') = \sum_i^n (x_i - x'_i)^2$$



### 3. Расстояние городских кварталов (манхэттенское расстояние)

Это расстояние является средним разностей по координатам. В большинстве случаев эта мера расстояния приводит к таким же результатам, как и для обычного расстояния Евклида. Однако для этой меры влияние отдельных больших разностей (выбросов) уменьшается (т.к. они не возводятся в квадрат). Формула для расчета манхэттенского расстояния:

$$\rho(x, x') = \sum_i^n |x_i - x'_i|$$

### 4. Расстояние Чебышева

Это расстояние может оказаться полезным, когда нужно определить два объекта как «различные», если они различаются по какой-либо одной координате. Расстояние Чебышева вычисляется по формуле:

$$\rho(x, x') = \max(|x_i - x'_i|)$$

## 5. Степенное расстояние

Применяется в случае, когда необходимо увеличить или уменьшить вес, относящийся к размерности, для которой соответствующие объекты сильно отличаются. Степенное расстояние вычисляется по следующей формуле:

$$\rho(x, x') = r \sqrt[r]{\sum_i^n (x_i - x'_i)^p},$$

где  $r$  и  $p$  – параметры, определяемые пользователем. Параметр  $p$  ответственен за постепенное взвешивание разностей по отдельным координатам, параметр  $r$  ответственен за прогрессивное взвешивание больших расстояний между объектами. Если оба параметра –  $r$  и  $p$  — равны двум, то это расстояние совпадает с расстоянием Евклида.

Хемингово расстояние:

$$p_H(x_i, x_j) = \sum_{k=1}^{l-1} |x_i^{1,2,\dots,k} - x_j^{1,2,\dots,k}|;$$

используется как мера различия объектов, задаваемых атрибутивными признаками.


Функции, которые используют в R для проведения иерархической кластеризации:

```
hclust(d, method = "complete")  
agnes(x, metric = "euclidean", stand = FALSE, method = "average")  
diana(x, metric = "euclidean", stand = FALSE)
```

где: d: матрица дистанций, полученная функцией dist() или как-то иначе;

method: метод кластеризации, определяемый одним из значений "ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median" или "centroid";

x: таблица данных для вычисления расстояний (наблюдения - по строкам, признаки - по столбцам);

 metric: метрика расстояний между наблюдениями, т.е. "euclidean" (Евклидово) или "manhattan" (Манхеттен);

stand = TRUE: осуществляется стандартизация каждой переменной таблицы (по столбцам).

1. Перечислите задачи, решаемые при помощи методов кластерного анализа.
2. Назовите две основные группы методов кластерного анализа и укажите их сходство и различие.
3. Какие меры сходства используются при проведении кластеризации?
4. Как оценивается качество полученного разбиения совокупности на кластеры в методе k-средних?
5. Какие команды в R позволяют проводить реализовывать кластеризацию?
6. Какие команды нужны для визуализации результатов кластерного анализа в R?

1. Воронцов К.В. [Алгоритмы кластеризации и многомерного шкалирования](#). Курс лекций. МГУ, 2007.
2. Jain A., Murty M., Flynn P. [Data Clustering: A Review](#). // ACM Computing Surveys. 1999. Vol. 31, no. 3.
3. Котов А., Красильников Н. [Кластеризация данных](#). 2006.
3. Мандель И. Д. Кластерный анализ. — М.: Финансы и Статистика, 1988.
4. Прикладная статистика: классификация и снижение размерности. / С.А. Айвазян, В.М. Бухштабер, И.С. Енюков, Л.Д. Мешалкин — М.: Финансы и статистика, 1989.
5. Информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных — [www.machinelearning.ru](http://www.machinelearning.ru)
6. Чубукова И.А. Курс лекций «Data Mining», Интернет-университет информационных технологий — [www.intuit.ru/departments/database/datamining](http://www.intuit.ru/departments/database/datamining)

1. Xu D., Tian Y. A comprehensive survey of clustering algorithms // Ann. Data Sci. – 2015. – V. 2, No 2. – P. 165–193. – doi: 10.1007/s40745-015-0040-1.
2. Sharan R., Shamir R. CLICK: A clustering algorithm with applications to gene expression analysis // Proc. Int. Conf. Intell. Syst. Mol. Biol. – AAAI Press, 2000. – P. 307–316.
3. Форд Л., Фалкерсон Д. Потоки в сетях. – М.: Наука, 1966. – 276 с. 4. Ford L.R. Jr., Fulkerson D.R., Maximal flow through a network // Can. J. Math. – 1956. – V. 8. – P. 399–404. – doi: 10.4153/CJM-1956-045-5.
4. Dinitz Y. Dinitz' algorithm: The original version and Even's version // Goldreich O., Rosenberg A.L., Selman A.L. (Eds.) Theoretical Computer Science. Lecture Notes in Computer Science, V. 3895. – Berlin, Heidelberg: Springer, 2006. – P. 218–240.
5. Edmonds J., Karp R.M. Theoretical improvements in algorithmic efficiency for network flow problems // J. Assoc. Comput. Mach. – 1972. – V. 19, No 2. – P. 248–264.
6. Сивоголовко Е. Методы оценки качества четкой кластеризации // Компьютерные
7. инструменты в образовании. – 2011. – Вып. 4. – С. 14–31.
8. Steinhaus H. Sur la division des corps materiels en parties // Bull. Acad. Polon. Sci., Cl. III. – 1956. – V. IV, No 12. – P. 801–804.
9. Lloyd S. Least square quantization in PCM // Trans. Inf. Theory. – 1982. – V. IT-28, No 2. – P. 129–137.
10. MacQueen J. Some methods for classification and analysis of multivariate observations // Proc. 5th Berkeley Symp. on Mathematical Statistics and Probability. – 1967. – P. 281–297.
11. Johnson S. Hierarchical clustering schemes // Psychometrika. – 1967. – V. 32, No 3. – P. 241–254. – doi: 10.1007/BF02289588.
12. Ward J.H. Hierarchical grouping to optimize an objective function // J. Am. Stat. Assoc. – 1963. – V. 58, No 301. – P. 236–244. – doi: 10.1080/01621459.1963.10500845.
13. Ester M., Kriegel H.-P., Sander J., Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise // Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD-96) – AAAI Press, 1996. – P. 226–231.
14. Franti P., Sieranoja S. Clustering basic benchmark. – URL: <http://cs.joensuu.fi/sipu/datasets/>.