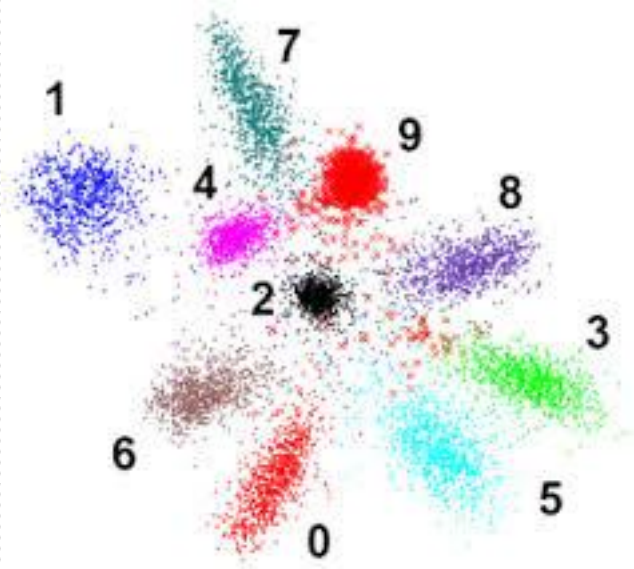
Кластеризация данных

ТЕХНОЛОГИИ ОБРАБОТКИ ИНФОРМАЦИИ



Ф.Филиппов, доцент кафедры ИУС

9000096@mail.ru

Разделы

- Основные определения
- Общая схема кластеризации
- Популярные алгоритмы
- Кластеризация в R

1

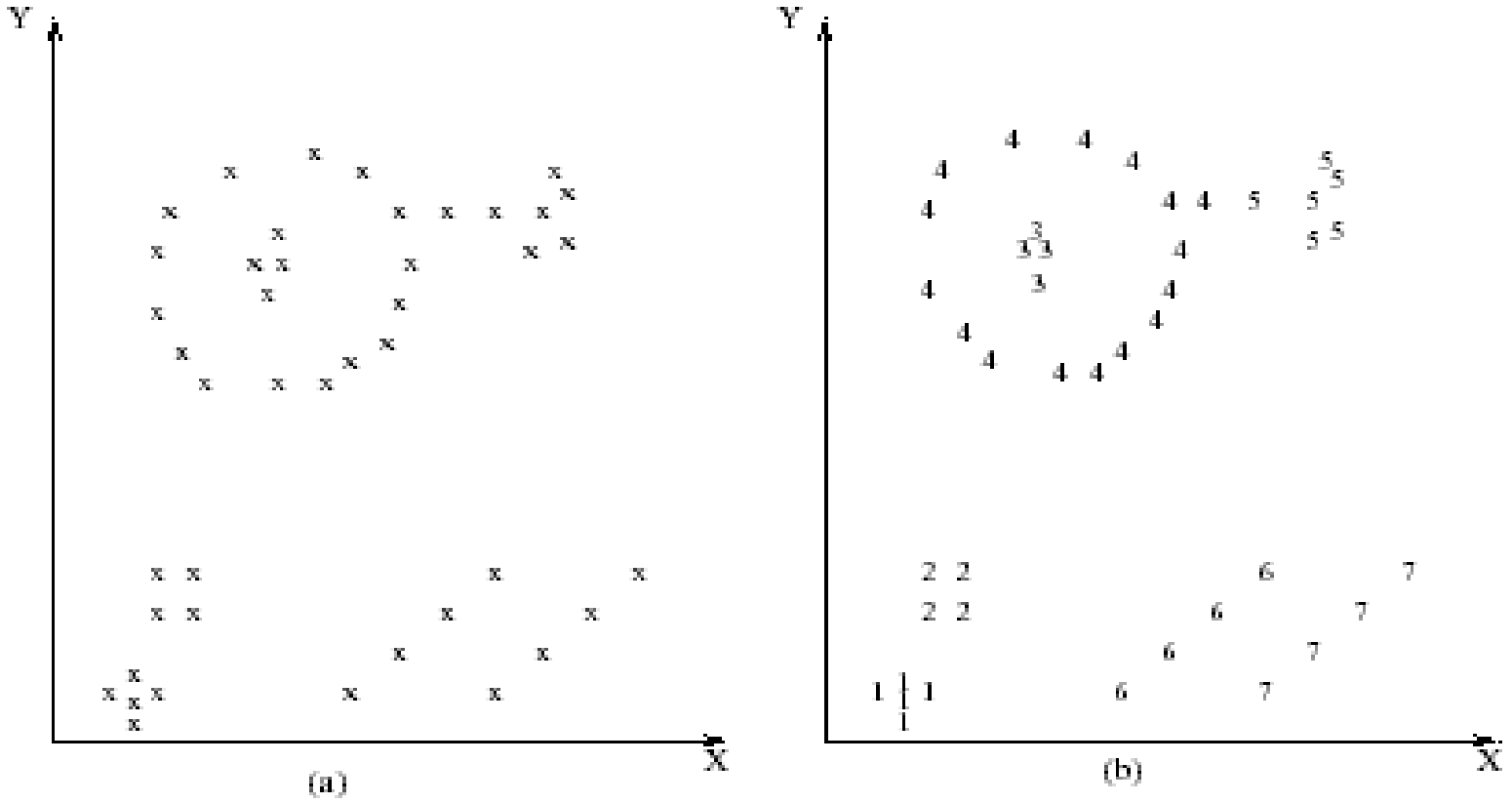
❑ Основные определения

Что такое кластеризация?

- **Кластеризация** – это автоматическое разбиение элементов некоторого множества (объекты, данные, векторы характеристик) на группы (кластеры) по принципу схожести.

Имя	Пол	Возраст	Город	Доход
Михаил	м	25	Москва	50000
Андрей	м	34	Санкт-Петербург	120000
Елена	ж	27	Нижний Новгород	15000
Наталья	ж	36	Санкт-Петербург	35000
Ярослав	м	53	Москва	180000
Ольга	ж	24	Москва	40000

Кластеризация (пример)



В чем отличие кластеризации и классификации?

- *Кластеризация* (unsupervised classification) разбивает множество объектов на группы, которые определяются только ее результатом.
- *Классификация* (supervised classification) относит каждый объект к одной из заранее определенных групп.

Зачем нужна кластеризация?

- Имеет много практических применений:
  - Анализ данных (Data mining);
  - Группировка и распознавание объектов;
  - Извлечение и поиск информации.
- Это важная форма абстракции данных.
- Это активно развивающаяся область технологии обработки данных.

Формальные определения

- Вектор характеристик (объект) $\boldsymbol{x}$ – единица данных для алгоритма кластеризации. Обычно это элемент d-мерного пространства: $\boldsymbol{x} = (x_1, \ldots, x_d)$.
- Характеристика (атрибут) x_i – скалярная компонента вектора $\boldsymbol{x}$.
- Размерность d – количество характеристик объекта $\boldsymbol{x}$.

Формальные определения

- Множество объектов $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ – набор входных данных. i-й объект из $\boldsymbol{X}$ определяется как $\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,d})$. Часто $\boldsymbol{X}$ представляют в виде *матрицы характеристик* размера n x d.
- Кластер – подмножество «близких друг к другу» объектов из $\boldsymbol{X}$.
- Расстояние $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ между объектами $\boldsymbol{x}_i$ и $\boldsymbol{x}_j$– результат применения выбранной метрики (или квази-метрики) в пространстве характеристик.

Постановка задачи

- Цель кластеризации – построить оптимальное разбиение объектов на группы - разбить *N* объектов на *k* кластеров
- Оптимальность разбиения можно определить как требование минимизации среднеквадратической ошибки разбиения:

$$e^2(X,k) = \sum_{j=1}^{k} \sum_{i=1}^{N} \left\| x_i^{(j)} - c_j \right\|^2$$

где c_j – «центр масс» кластера j

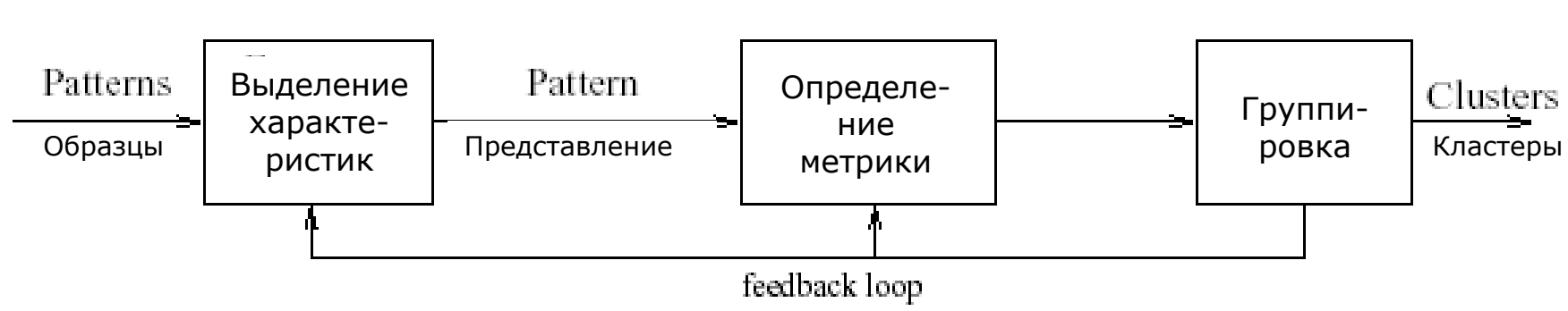
2

❑ Общая схема кластеризации

Общая схема кластеризации

Выделение характеристик
Определение (выбор) метрики
Разбиение объектов на группы
Представление результатов

Образцы



Выделение характеристик

- Выбор свойств, характеризующих объекты:
  - количественные характеристики (координаты, интервалы…);
  - качественные характеристики (цвет, статус, состав…).
- Уменьшение размерности пространства, нормализация характеристик.
- Представление объектов в виде характеристических векторов.

Выбор метрики

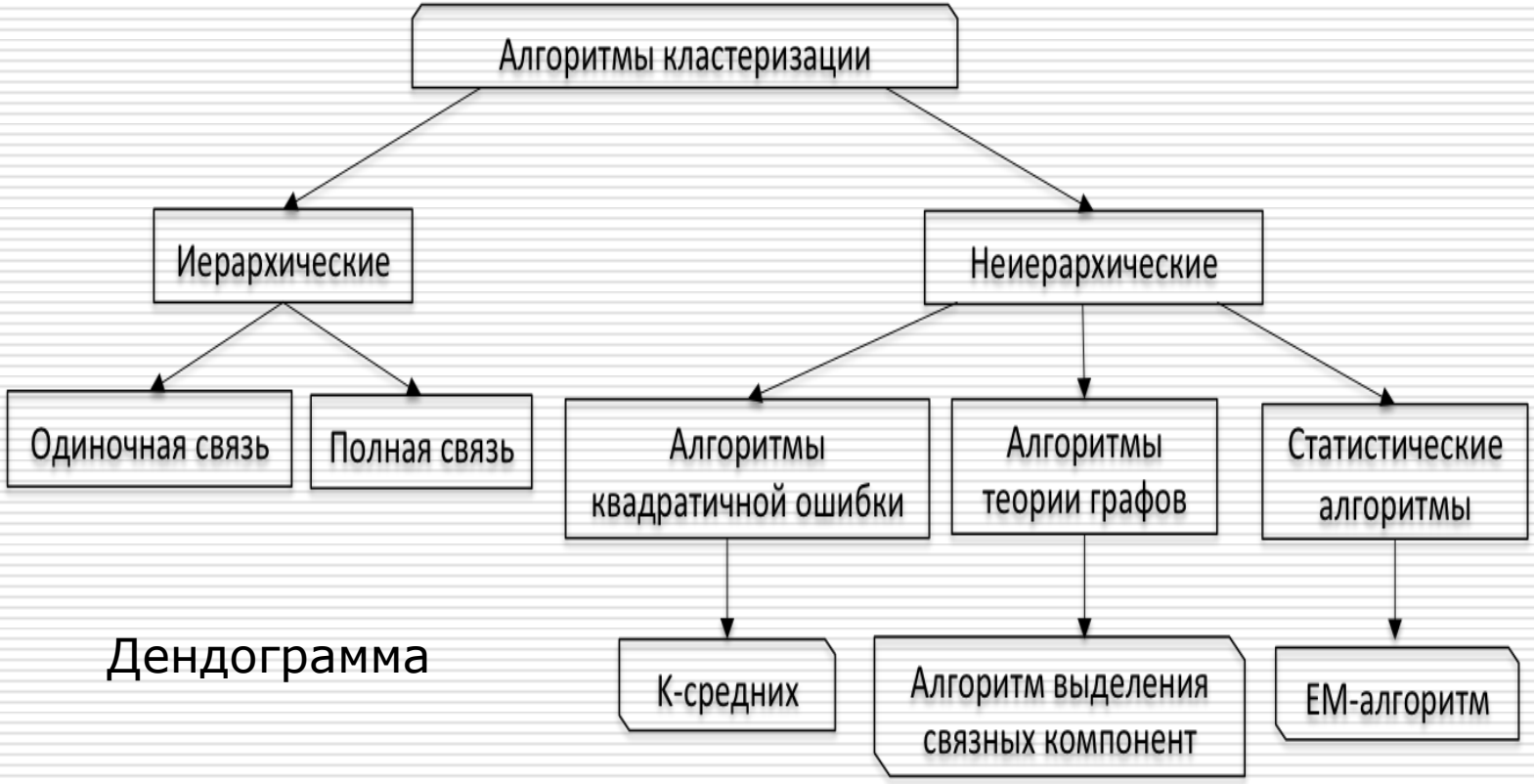
- Метрика выбирается в зависимости от:
  - пространства, где расположены объекты;
  - неявных характеристик кластеров.
- Если все координаты объекта непрерывны и вещественны, то используется классическая метрика Евклида:

$$d_2\left(x_i, x_j\right) = \left(\sum_{k=1}^{d} \left(x_{i,k} - x_{j,k} \right)^2 \right)^{1/2} = \left\| x_i - x_j \right\|_2$$

3

Популярные алгоритмы

Алгоритмы кластеризации (схема)



Дендограмма

Популярные алгоритмы кластеризации

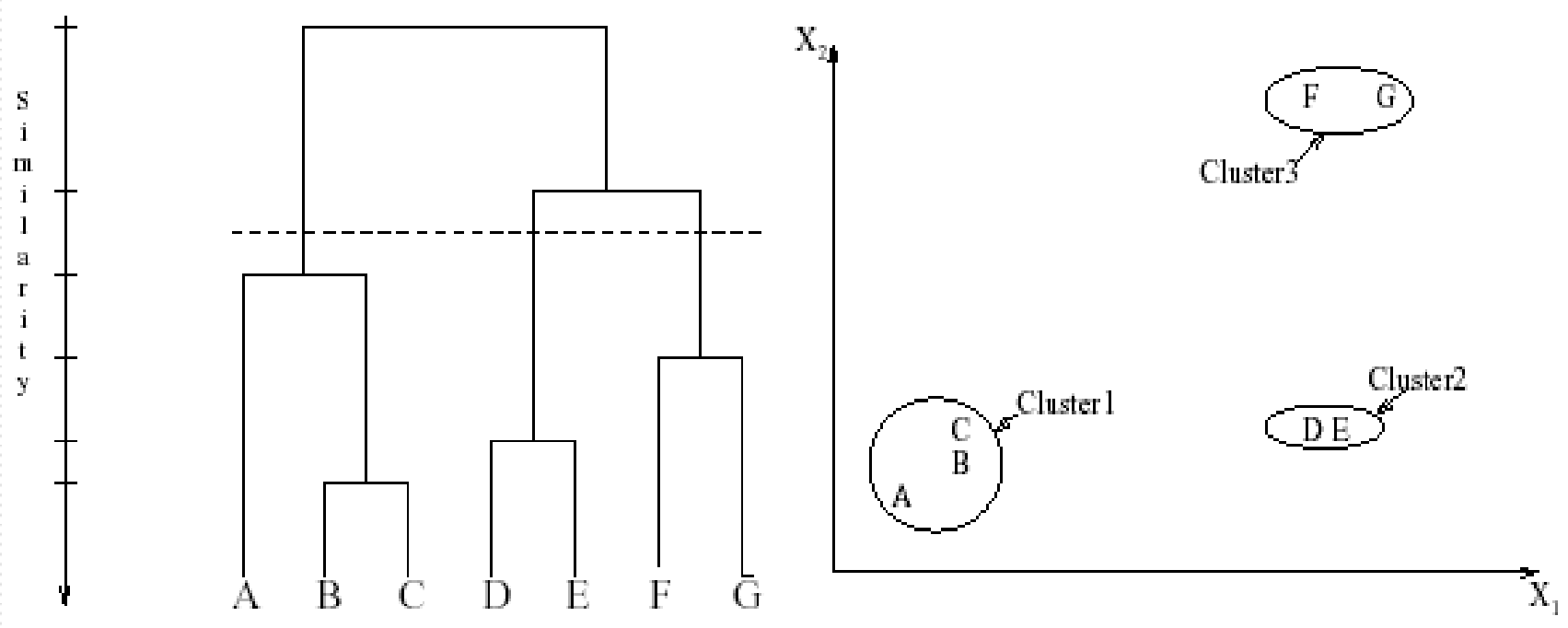
1. Иерархические алгоритмы
2. Минимальное покрывающее дерево
3. Алгоритм *k*-средних (*k-Means*)
4. Метод ближайшего соседа

1. Применение нейронных сетей

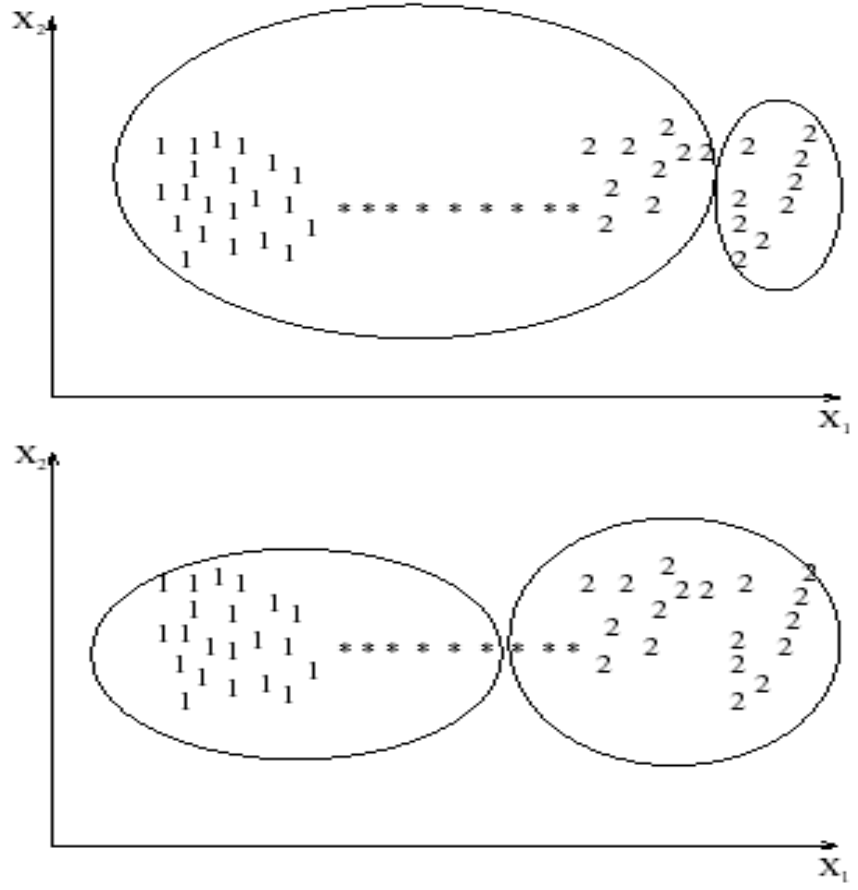
① Иерархические алгоритмы

- Результатом работы является *дендограмма* (иерархия), позволяющая разбить исходное множество объектов на любое число кластеров.
- Два наиболее популярных алгоритма, оба строят разбиение «снизу-вверх»:
  - Single-link – на каждом шаге объединяет два кластера с наименьшим расстоянием между двумя *наиболее близкими* представителями;
  - Complete-link – объединяет кластеры с наименьшим расстоянием между двумя *наиболее удаленными* представителями.

Пример Single-link

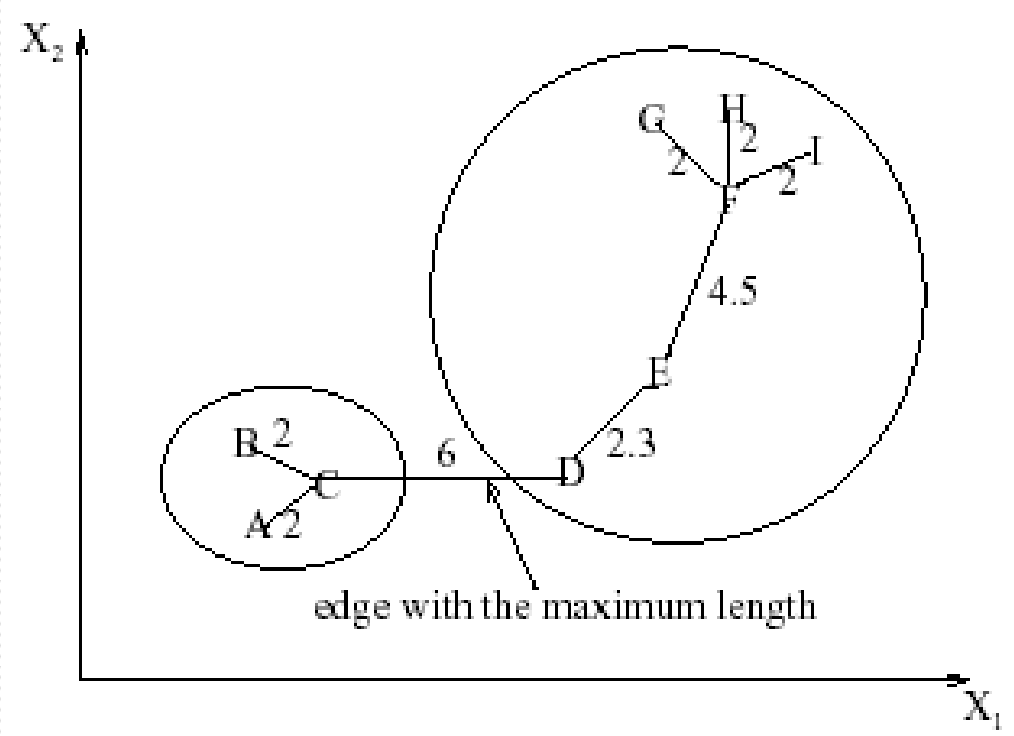


Сравнение
Single-link и Complete-link



② Минимальное покрывающее дерево

- Позволяет производить иерархическую кластеризацию «сверху-вниз»:

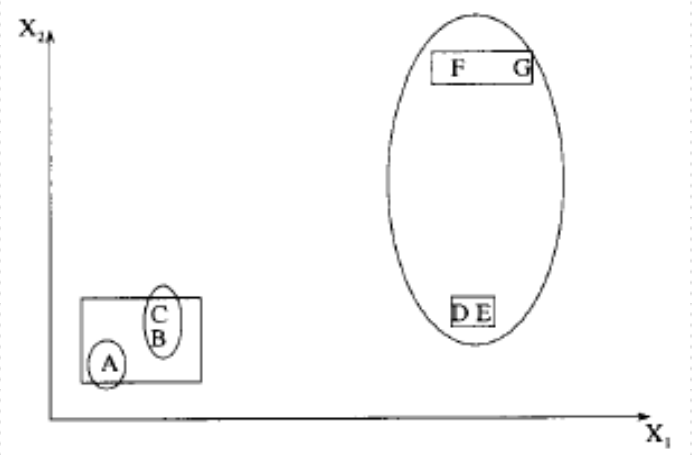


③ *k-Means* алгоритм

1. Случайно выбрать *k* точек, являющихся начальными «центрами масс» кластеров (любые *k* из *n* объектов, или вообще *k* случайных точек).
2. Отнести каждый объект к кластеру с ближайшим «центром масс».
3. Пересчитать «центры масс» кластеров согласно текущему членству.
4. Если критерий остановки алгоритма не удовлетворен, вернуться к шагу 2.

k-Means алгоритм (продолжение)

- В качестве критерия остановки обычно выбирают один из двух:
  - Отсутствие перехода объектов из кластера в кластер на шаге 2;
  - Минимальное изменение среднеквадратической ошибки.
- Алгоритм чувствителен к начальному выбору «центров масс».



④ Метод ближайшего соседа

- Один из старейших (1978), простейших и наименее оптимальных алгоритмов:

Пока существуют объекты вне кластеров {

- Для каждого такого объекта выбрать ближайшего соседа, кластер которого определен, и если расстояние до этого соседа меньше порога – отнести его в тот же кластер, иначе можно создать новый;
- Увеличить порог при необходимости;

}

⑤ Применение нейронных сетей

- Нейронные сети (НС) легко работают в распределенных системах в силу своей природы.
- Настройка весовых коэффициентов НС помогает сделать выбор характеристик (этап 1 кластеризации) менее субъективным.
- Кластеризация с применением самоорганизующихся карт Кохонена эквивалентна алгоритму *k-Means*.

Рассмотрим в теме «Нейронные сети»

Понижение размерности наблюдаемых данных

- Зачастую, наблюдаемые данные могут обладать высокой размерностью, но в действительности быть функцией всего нескольких скрытых (латентных) переменных
- Задачей понижения размерности является некоторое преобразование исходного пространства в пространство более низкой размерности без существенной потери информативности данных

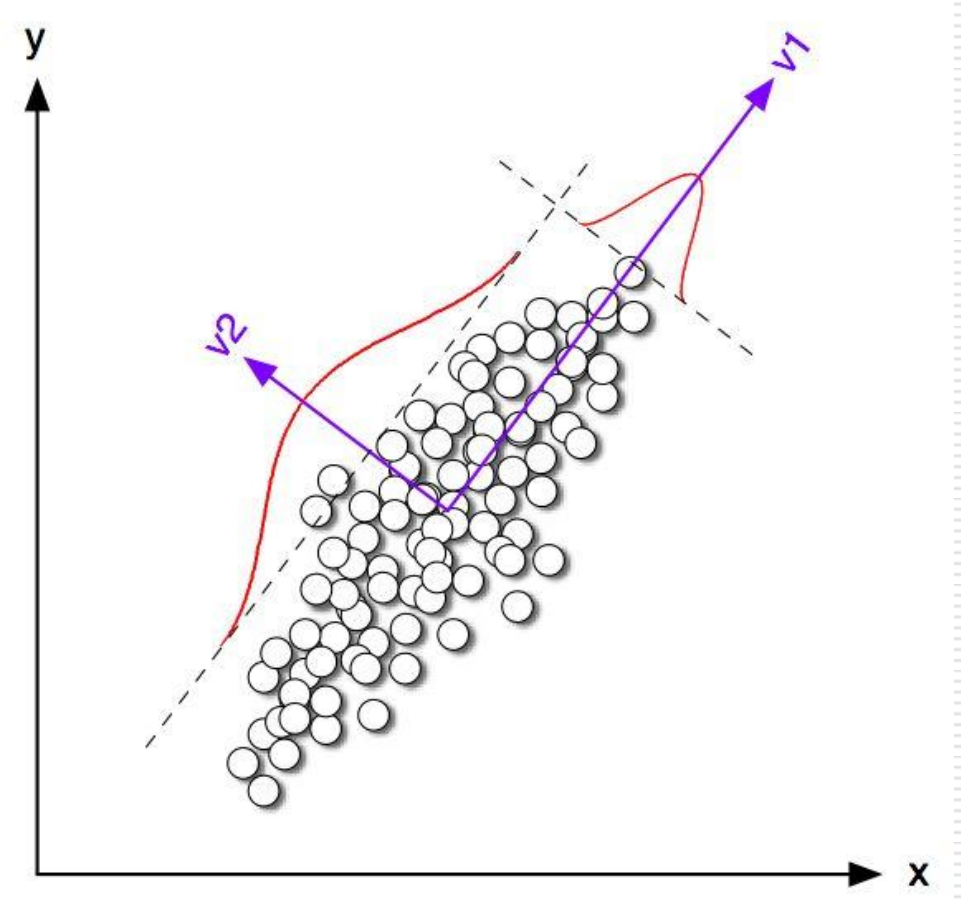
Метод главных компонент

- Пусть имеется набор наблюдений $X^l = \{x_1, \ldots, x_l\}, X \in R^d$
- Будем строить новый базис в пространстве R^d, таким образом что:
  - Центр координат совпадает с мат. ожиданием наблюдений (выборочным средним)
  - Первый вектор направлен таким образом, что дисперсия вдоль него была максимальной
  - Каждый последующий вектор ортогонален предыдущим и направлен по направлению максимальной дисперсии

Иллюстрация

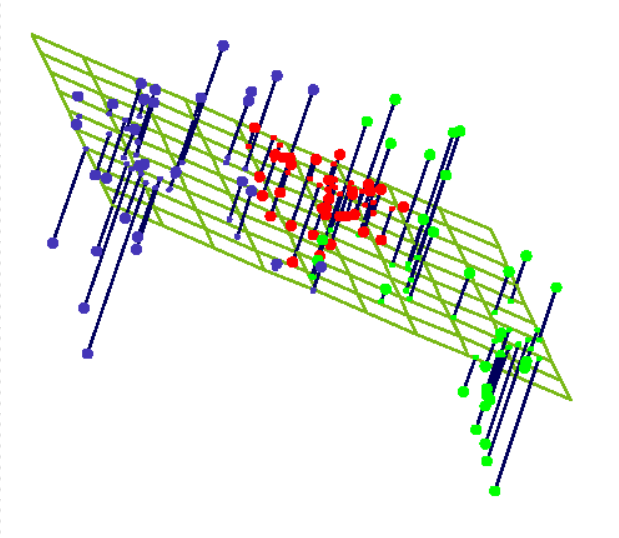
- Убирая базисные вектора с малыми значениями мы можем сократить размерность без существенной потери информации

***Вопрос: Почему это так?**



Метод главных компонент
Связь с линейной аппроксимацией

- Если рассмотреть систему проекций данных на первые $n \pounds d$ главных компонент, то мы получим систему наилучших линейных приближений данных (в смысле среднеквадратичного отклонения)

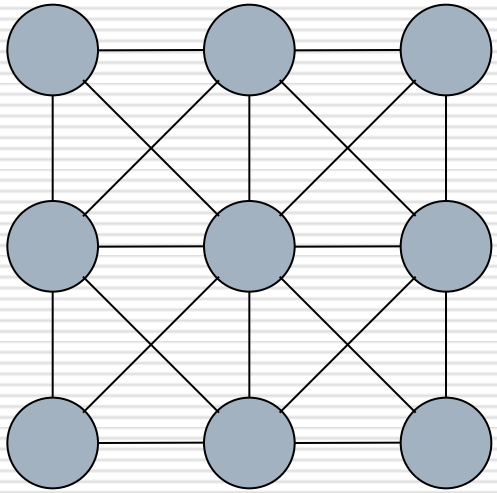


Самоорганизующиеся карты SOM (Карты Кохонена)

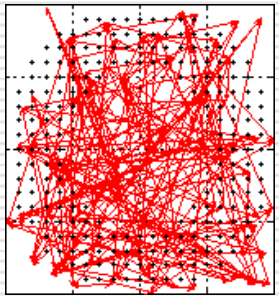
- Основная идея
  - вписать в данные сетку низкой размерности, и анализировать ее, вместо самих данных

SOM (Карты Кохонена) Модель сетки

- Матрица узлов $M^{KL} = (m_{kl})$
- Соседство - 4 или 8 связность
- Каждому узлу соответствует точка в исходном пространстве $m_{kl} \in R^d$



SOM (Карты Кохонена) Алгоритм построения

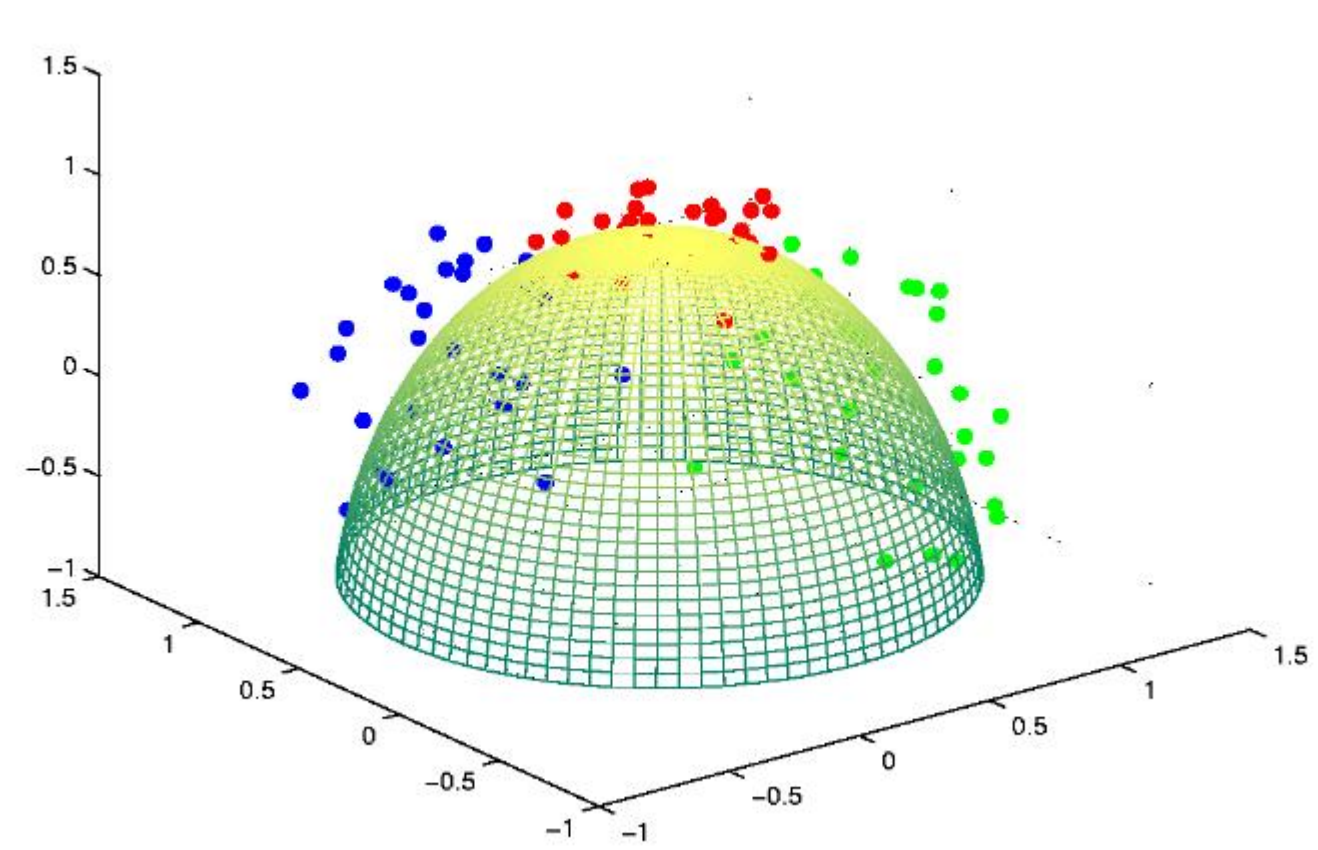


- Проинициализируем $\left(m_{ij}\right)$ случайными значениями
- Далее, в случайном порядке будем предъявлять наблюдения и для каждого:
  - Вычисляем ближайший узел
  - Выберем множество соседей узла, такое что расстояние на сетке между ними меньше r
  - Для некоторого множества соседей узла, включая сам узел, изменяем их положения согласно:

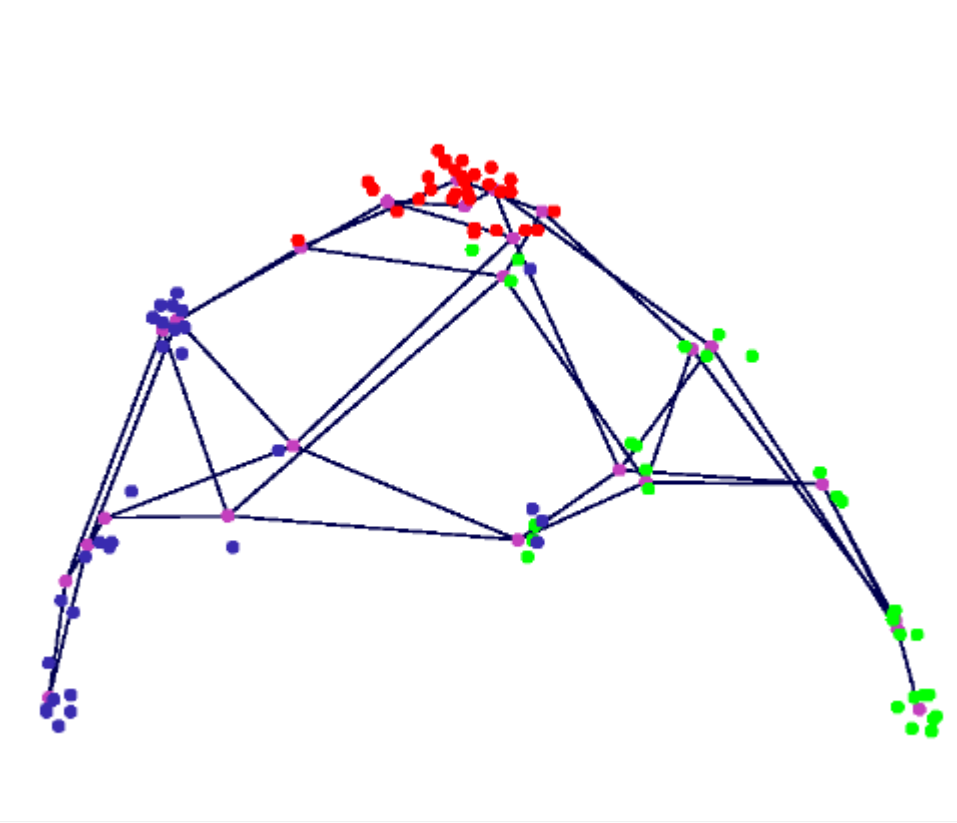
$$m_{kl} \leftarrow m_{kl} + \alpha\left(x_i - m_{kl}\right)$$

  - Повторяем процедуру уменьшая r и α пока сеть не стабилизируется

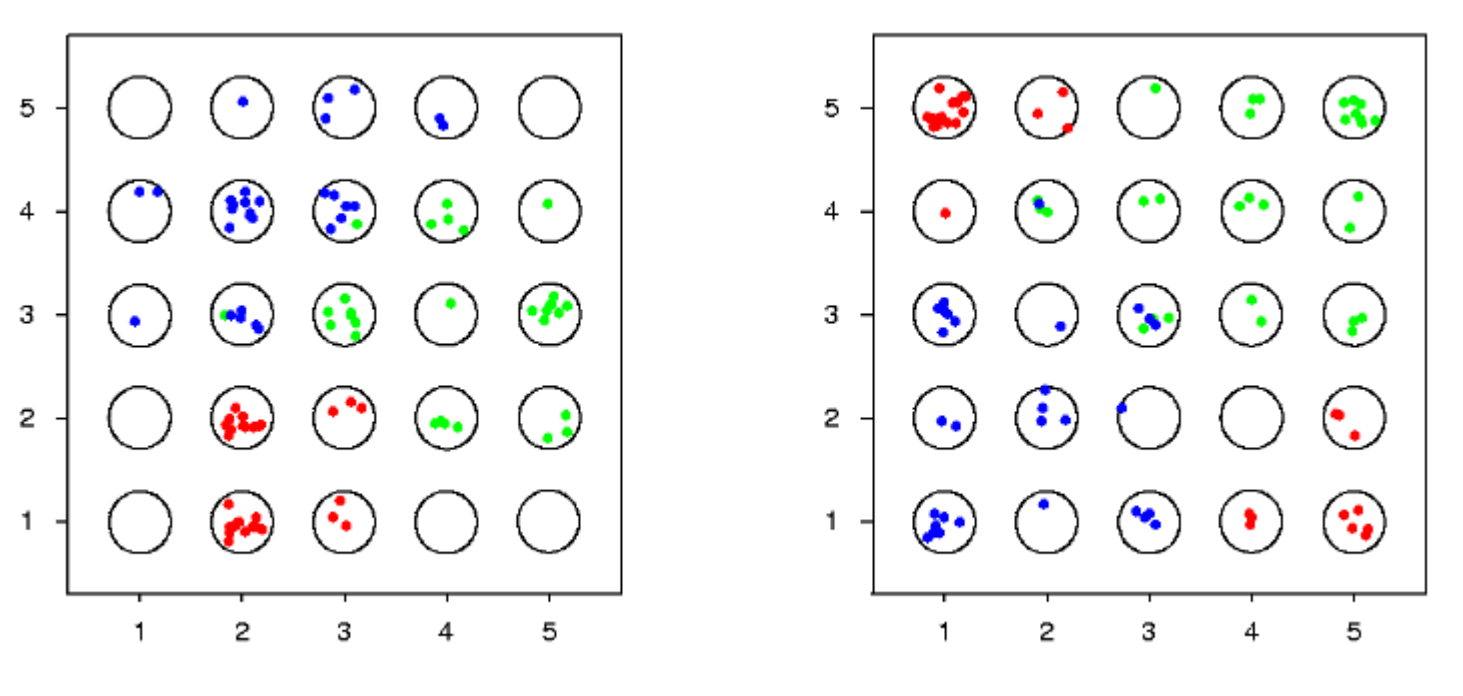
SOM (Карты Кохонена)
Иллюстрация: исходные данные



SOM (Карты Кохенена)
Иллюстрация: сетка



SOM (Карты Кохенена)
Иллюстрация: проекции на матрицу



SOM (Карты Кохенена) Практическое использование

- Данные представляют некоторую поверхность, требуется сократить размерность
- Хорошо подходят для последующей кластеризации
- Могут работать «online»
- В случае слишком сложных данных не информативны

Какой алгоритм выбрать?

- *k-Means* быстро работает и прост в реализации, но результат зависит от начального распределения.
- Иерархические алгоритмы дают оптимальное разбиение на кластеры, но их трудоемкость квадратична.
- На практике лучше всего зарекомендовали себя гибридные подходы, где шлифовка кластеров выполняется методом *k-Means*, а их первоначальное разбиение – одним из более сильных методов.

Кластеризация больших объемов данных

- Обычно используют *k*-Means или его гибридные модификации.
- Если множество объектов не помещается в основную память, можно:
  - проводить кластеризацию по принципу «разделяй и властвуй»;
  - использовать потоковые (on-line) алгоритмы (например, leader, модификация метода ближайшего соседа);
  - использовать параллельные вычисления.

4

Кластеризация в R

RStudio

Untitled1*

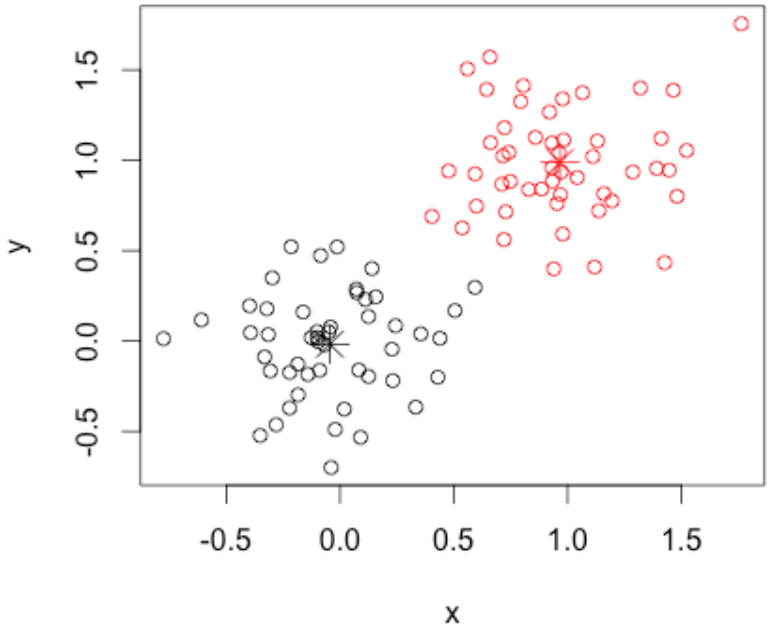
```
require(graphics)
x <- rbind(matrix(rnorm(100, sd = 0.3), ncol = 2),
           matrix(rnorm(100, mean = 1, sd = 0.3), ncol = 2))
colnames(x) <- c("x", "y")
(cl <- kmeans(x, 2))
plot(x, col = cl$cluster)
points(cl$centers, col = 1:2, pch = 8, cex = 2)
```

Console ~/Desktop/

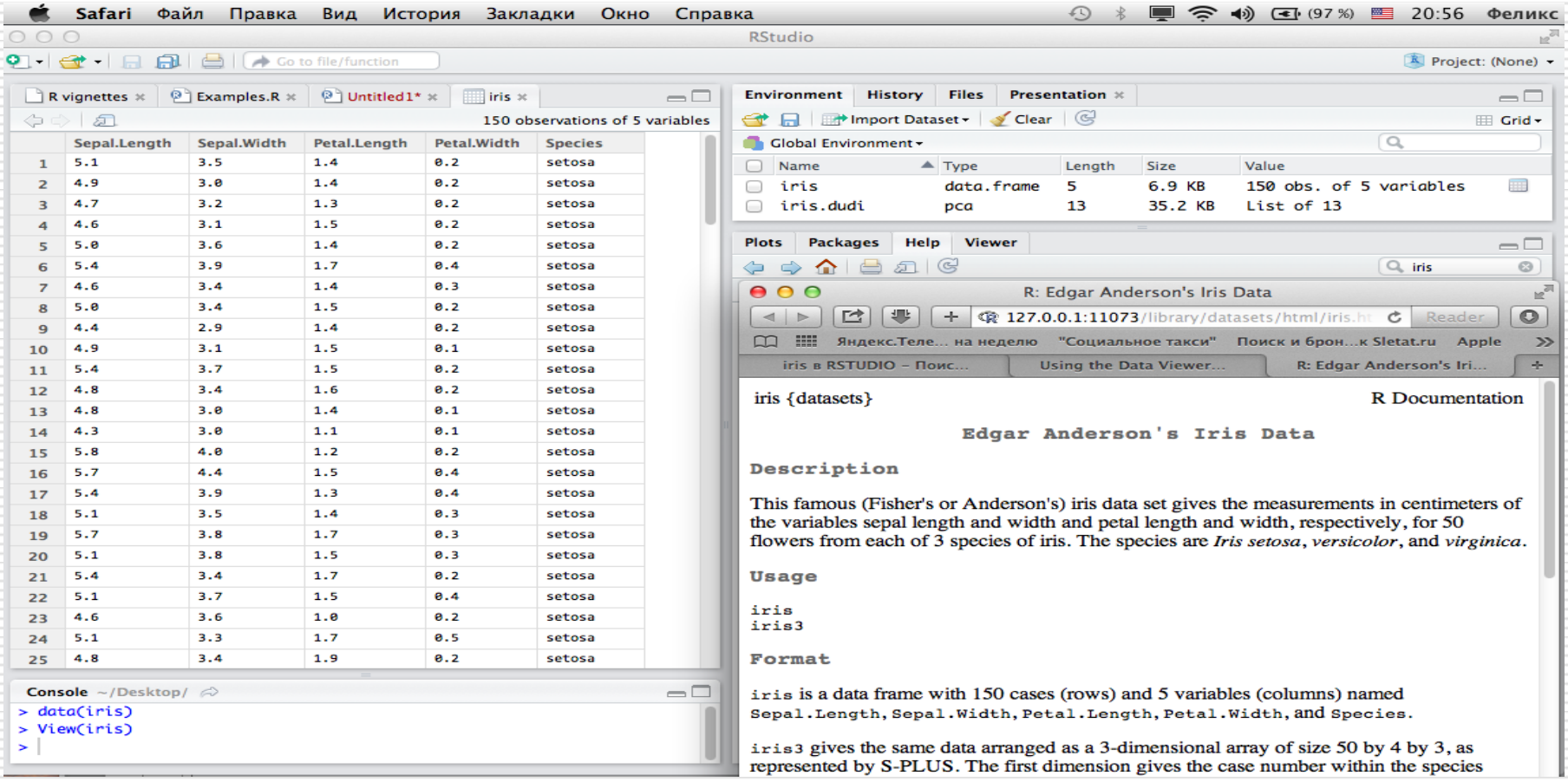
```
> (cl <- kmeans(x, 2))
K-means clustering with 2 clusters of sizes 49, 51

Cluster means:
           x          y
1 -0.04465995 -0.01750318
2  0.96715089  0.98904277

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1
 [35] 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [69] 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

Within cluster sum of squares by cluster:
[1] 7.587885 9.211771
 (between_SS / total_SS =  75.2 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"
[5] "tot.withinss" "betweenss"    "size"         "iter"
[9] "ifault"
> plot(x, col = cl$cluster)
> points(cl$centers, col = 1:2, pch = 8, cex = 2)
```

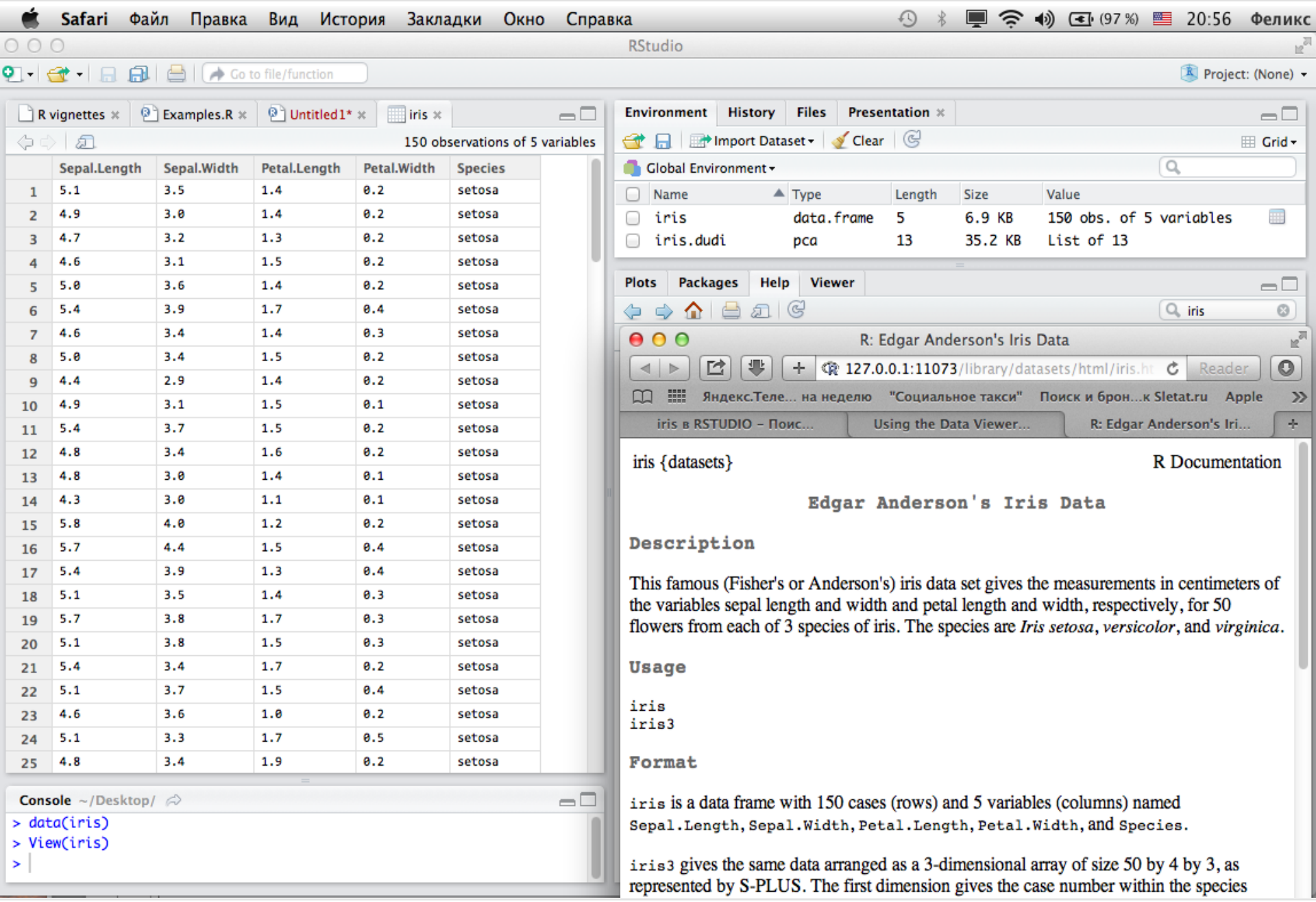
Environment

Name	Type	Length	Size	Value
cl	kmeans	9	2.5 KB	List of 9
x	matrix	200	2.1 KB	num [1:100, 1:2] …

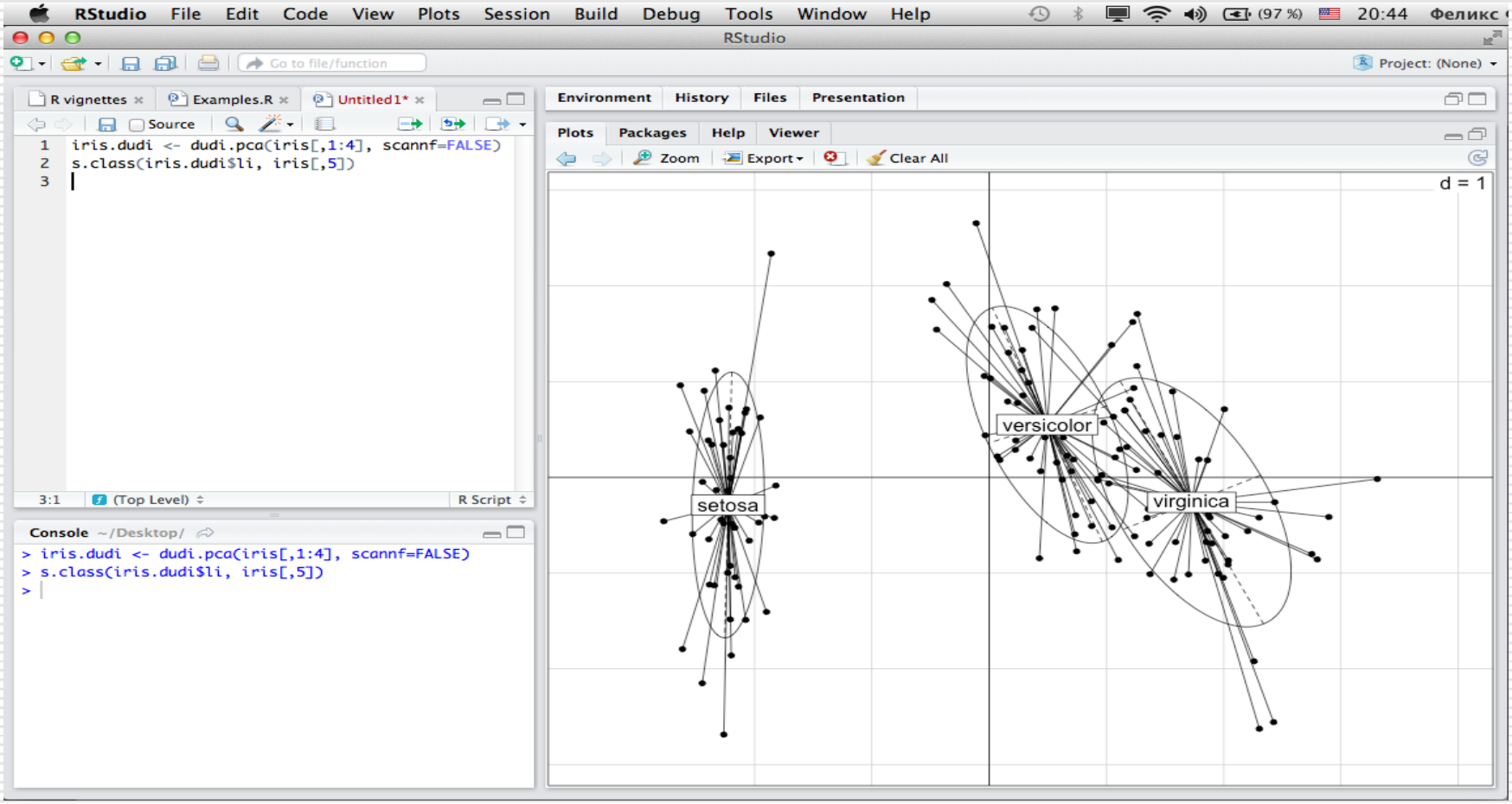


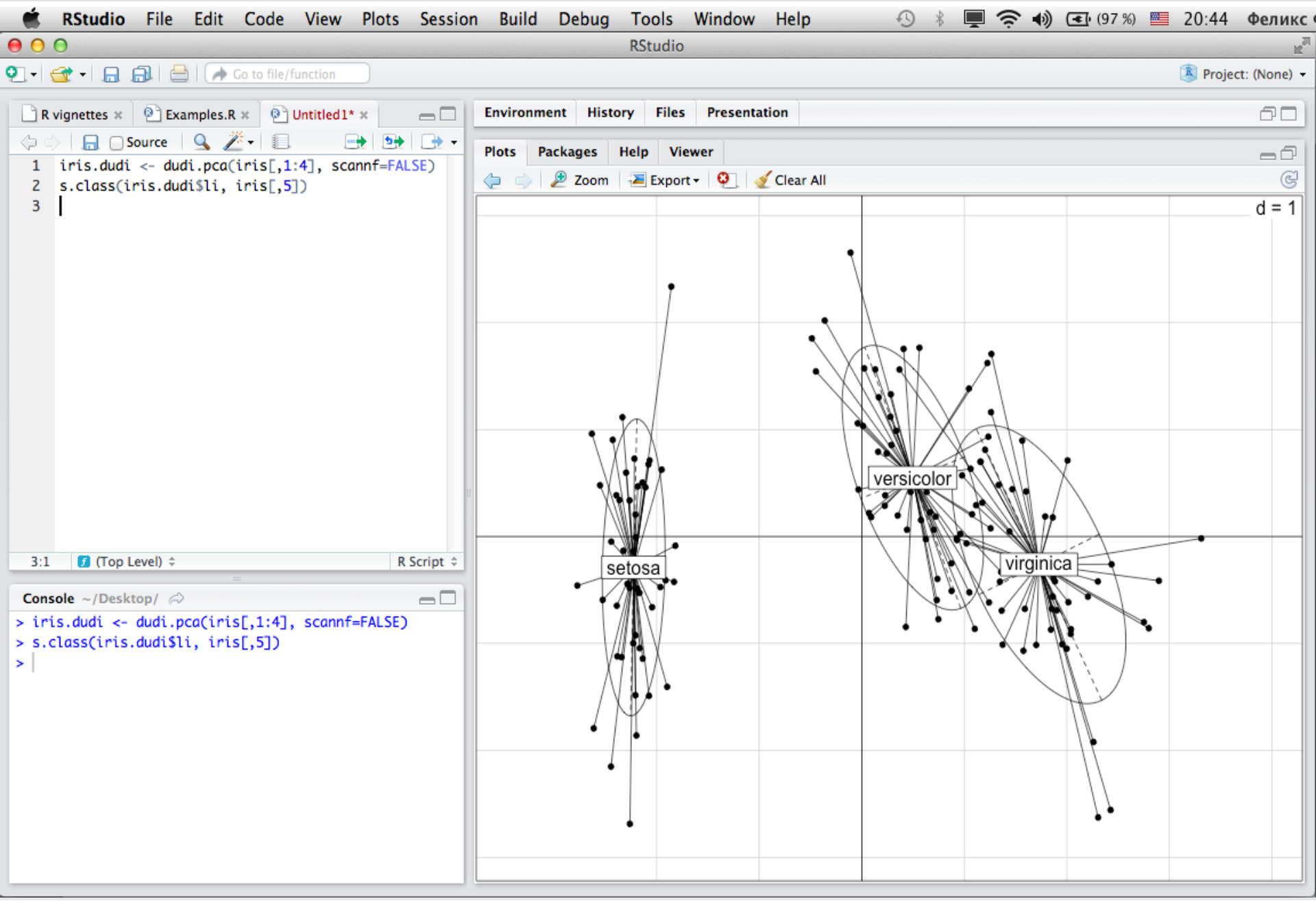
Тренировочные данные ***iris***




Safari Файл Правка Вид История Закладки Окно Справка
20:56 Феликс
RStudio
Project: (None)
R vignettes
Examples.R
Untitled1*
iris
150 observations of 5 variables
Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1 5.1 3.5 1.4 0.2 setosa
2 4.9 3.0 1.4 0.2 setosa
3 4.7 3.2 1.3 0.2 setosa
4 4.6 3.1 1.5 0.2 setosa
5 5.0 3.6 1.4 0.2 setosa
6 5.4 3.9 1.7 0.4 setosa
7 4.6 3.4 1.4 0.3 setosa
8 5.0 3.4 1.5 0.2 setosa
9 4.4 2.9 1.4 0.2 setosa
10 4.9 3.1 1.5 0.1 setosa
11 5.4 3.7 1.5 0.2 setosa
12 4.8 3.4 1.6 0.2 setosa
13 4.8 3.0 1.4 0.1 setosa
14 4.3 3.0 1.1 0.1 setosa
15 5.8 4.0 1.2 0.2 setosa
16 5.7 4.4 1.5 0.4 setosa
17 5.4 3.9 1.3 0.4 setosa
18 5.1 3.5 1.4 0.3 setosa
19 5.7 3.8 1.7 0.3 setosa
20 5.1 3.8 1.5 0.3 setosa
21 5.4 3.4 1.7 0.2 setosa
22 5.1 3.7 1.5 0.4 setosa
23 4.6 3.6 1.0 0.2 setosa
24 5.1 3.3 1.7 0.5 setosa
25 4.8 3.4 1.9 0.2 setosa
Console ~/Desktop/
> data(iris)
> View(iris)
>
Environment History Files Presentation
Import Dataset Clear Grid
Global Environment
Name Type Length Size Value
iris data.frame 5 6.9 KB 150 obs. of 5 variables
iris.dudi pca 13 35.2 KB List of 13
Plots Packages Help Viewer
iris
R: Edgar Anderson's Iris Data
127.0.0.1:11073/library/datasets/html/iris.ht
Reader
Яндекс.Теле... на неделю "Социальное такси" Поиск и брон...к Sletat.ru Apple
iris в RSTUDIO – Поис... Using the Data Viewer... R: Edgar Anderson's Iri...
iris {datasets} R Documentation
Edgar Anderson's Iris Data
Description
This famous (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are Iris setosa, versicolor, and virginica.
Usage
iris
iris3
Format
iris is a data frame with 150 cases (rows) and 5 variables (columns) named Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, and Species.
iris3 gives the same data arranged as a 3-dimensional array of size 50 by 4 by 3, as represented by S-PLUS. The first dimension gives the case number within the species

Кластеризация данных ***iris***




RStudio File Edit Code View Plots Session Build Debug Tools Window Help
20:44
RStudio
Project: (None)
R vignettes
Examples.R
Untitled1*
Source
1 iris.dudi <- dudi.pca(iris[,1:4], scannf=FALSE)
2 s.class(iris.dudi$li, iris[,5])
3
3:1
(Top Level)
R Script
Console ~/Desktop/
> iris.dudi <- dudi.pca(iris[,1:4], scannf=FALSE)
> s.class(iris.dudi$li, iris[,5])
>
Environment
History
Files
Presentation
Plots
Packages
Help
Viewer
Zoom
Export
Clear All
d = 1
setosa
versicolor
virginica

Кластеризация реальных данных (пример 1)

Возьмем котировки 22 российских компаний из разных областей, за два года с периодом один день:

```
DATE,AFLT,GAZP,KMAZ,LKOH,VTBR,URKA,TRNFP,SIBN,SBER,ROSN,RASP,PMTL,PLZL,OGK1,MMBM,GMKN,OGK2,OGK5,AVAZ,MTSI,SNGS,VZRZ
20090428,32.32,143,25.39,1457,0.0307,77.74,12480,85.61,26.1,163.67,51.2,201.7,1382,0.36,705,2635,0.434,0.909,10.93,161.4,22.851,439
20090429,32.51,149,30.8,1509,0.0317,79.35,13599.78,87.96,27.11,177.21,51.48,204.49,1389.44,0.409,710,2687.84,0.495,0.972,11.06,166.5,24.038,429
20090430,33.96,147.82,30.23,1481.99,0.0315,79.4,13538.51,88.74,27.8,176.99,53.9,202.76,1373.89,0.411,717.1,2775,0.491,0.974,10.753,164.64,23.5,426.99
20090504,34.98,155,34.87,1547.79,0.033,82.19,15044,93.38,29.28,181.55,57.49,204.9,1376.8,0.429,728,2898,0.493,1.029,11.001,173.09,24.274,434
20090505,35,161.3,33.7,1504,0.0332,82.5,14500,92.07,28.87,177.82,57.1,217,1419,0.419,709,2903,0.476,1.007,11.467,173.48,24,427.64
20090506,35.19,169.91,34.37,1565.97,0.0339,92.2,14240,93.99,29.38,181.74,58,212.54,1405.07,0.457,708,3038.99,0.522,1.132,11.34,175.52,23.959,457.99
20090507,35.49,171.61,36.36,1589,0.0361,96.8,14299.99,95.2,31.15,183.72,57.65,215.97,1407.5,0.448,707,3162,0.51,1.104,11.205,175.8,24.8,499
20090508,35.49,173.3,35.96,1631.51,0.0418,97.5,14516,96.49,32.27,181.6,59.23,214,1404.01,0.447,721.99,3231.56,0.515,1.149,11.303,181.6,25.3,501
20090512,36.99,179.05,37,1613.87,0.046,115.06,15635,99.7,33.65,187,60.79,216.4,1400,0.453,724.22,3384.99,0.535,1.13,11.403,175.01,26.2,535
```

Т.к. цены разных акций отличаются очень сильно, и абсолютная величина цены нам совершенно не интересна, а интересно относительное изменение, приведем все цены к общему знаменателю - вместо самой цены будем считать логарифмическое изменение цены:

$$X_i = log(C_i) - log(C_{i-1}) ,$$

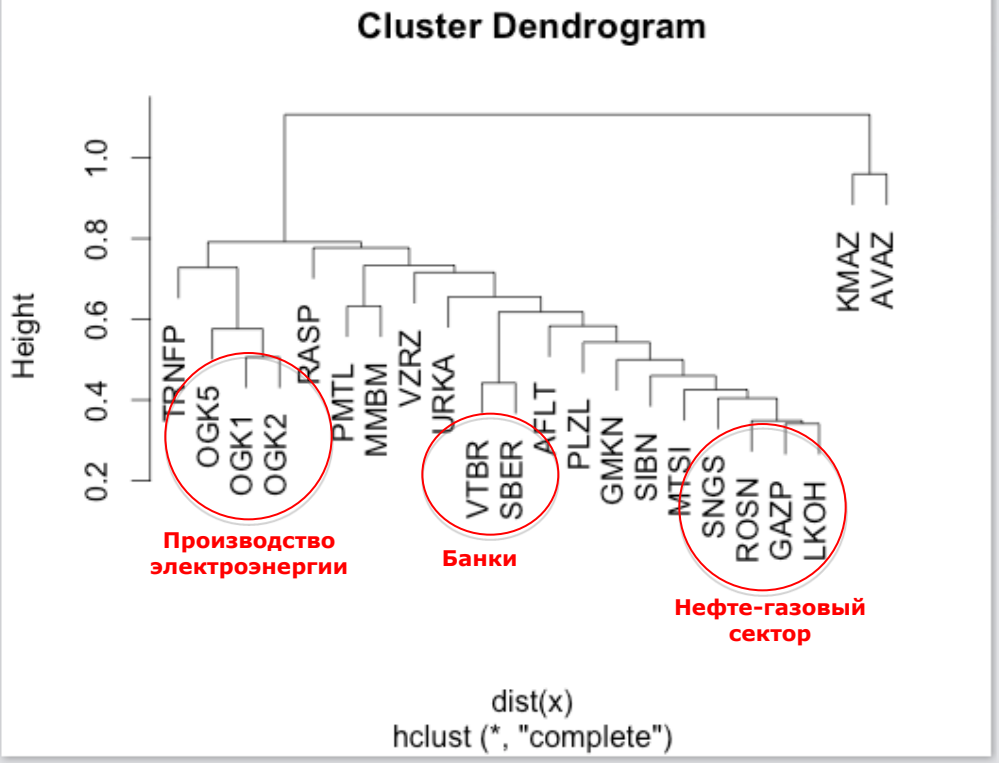
где C_i – цена акции в i-ый день.

https://github.com/Edunov/Samples/blob/master/stocks.csv

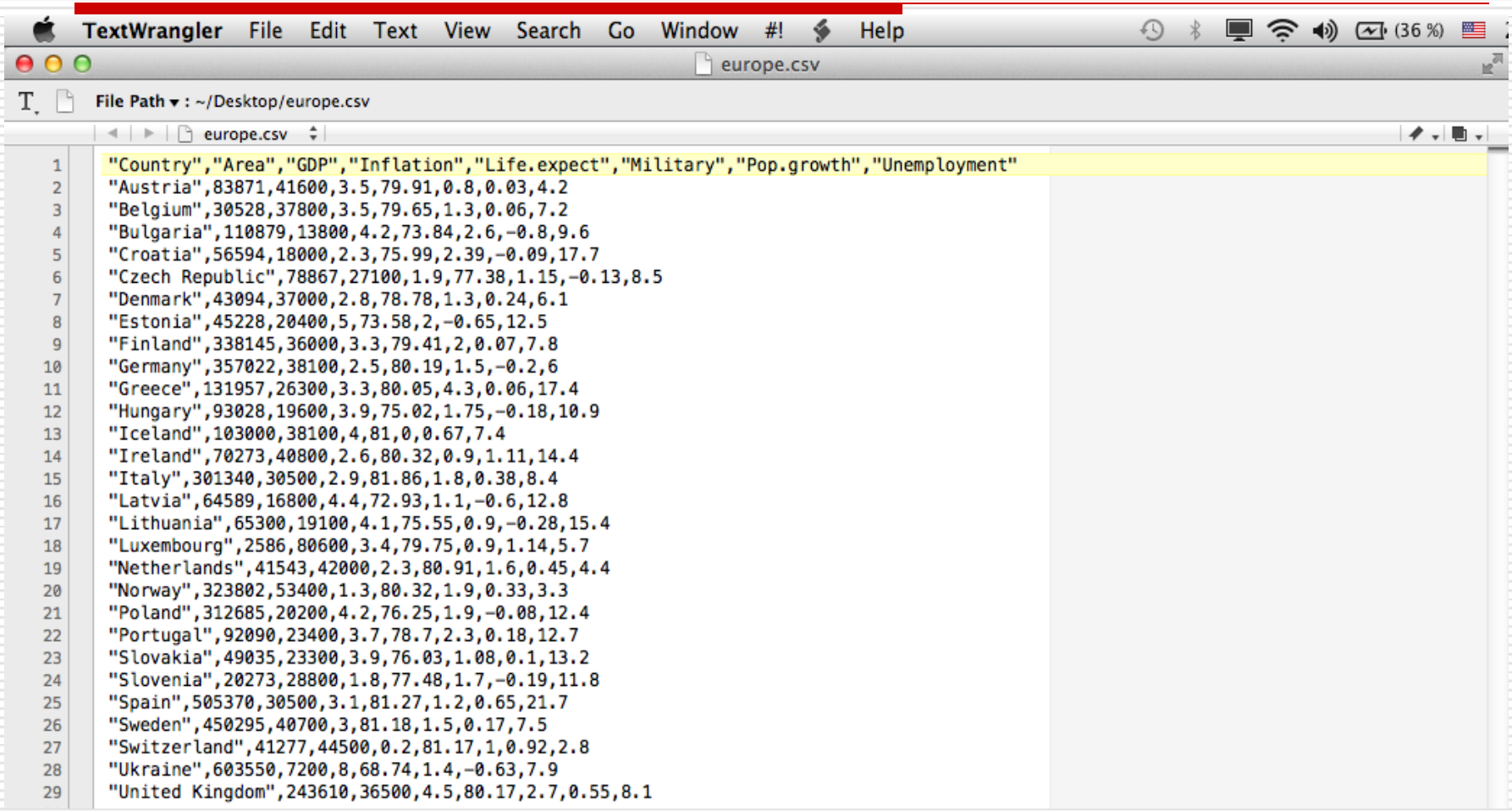
RStudio

```
x <- read.table('stock.csv', sep=',', header=TRUE)
x <- log(x)              # логарифмируем цены
x <- x[-1]               # отбросим колонку с датой
x <- apply(x, 2, diff)   # считаем разницу между элементами
x <- t(x)                # транспонируем таблицу
hc <- hclust(dist(x))    # вычисляем иерархический кластер
plot(hc)                 # строим график
```

```
> x <- read.table('stock.csv', sep=',', header=TRUE)
> x <- log(x)              # логарифмируем цены
> x <- x[-1]               # отбросим колонку с датой
> x <- apply(x, 2, diff)   # считаем разницу между элементами
> x <- t(x)                # транспонируем таблицу
> hc <- hclust(dist(x))    # вычисляем иерархический кластер
> plot(hc)                 # строим график
>
```



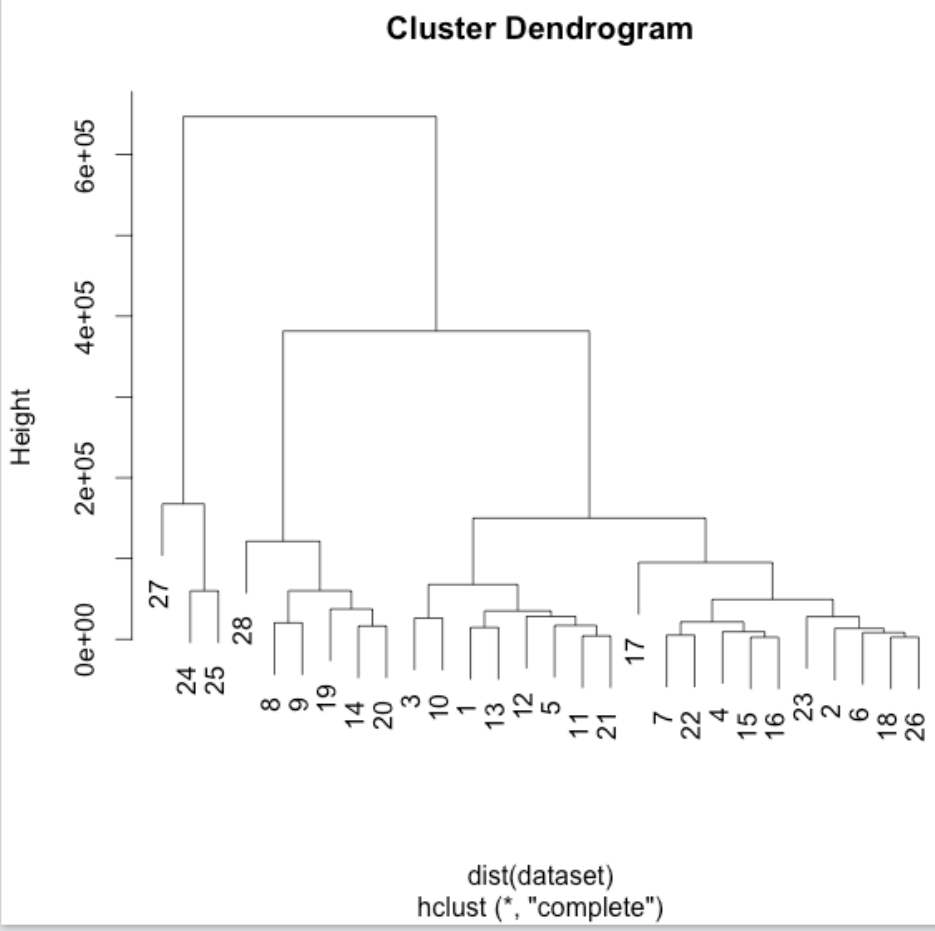
Кластеризация реальных данных (пример 2)



http://www.instantr.com/2013/02/12/performing-a-cluster-analysis-in-r/

```
dataset <- read.csv(file="europe.csv",head=TRUE,sep=",")
modelname<-hclust(dist(dataset))
plot(modelname)
plot(modelname, labels=dataset$Country)
rect.hclust(modelname, 12)
rect.hclust(modelname, h=2e+05)
```

```
> dataset <- read.csv(file="europe.csv",head=TRUE,sep=",")
> modelname<-hclust(dist(dataset))
Предупреждение:
В dist(dataset) : в результате преобразования созданы NA
> plot(modelname)
>
```



RStudio File Edit Code View Plots Session Build Debug Tools Window Help (32 %) 22:44 Феликс

RStudio

Open an existing file (⌘O)

Моя новая презентация.Rpres

Source on Save | Run | Source

```
1  dataset <- read.csv(file="europe.csv",head=TRUE,sep=",")
2  modelname<-hclust(dist(dataset))
3  plot(modelname)
4  plot(modelname, labels=dataset$Country)
5  rect.hclust(modelname, 12)
6  rect.hclust(modelname, h=2e+05)
7
8
```

5:1 (Top Level) R Script

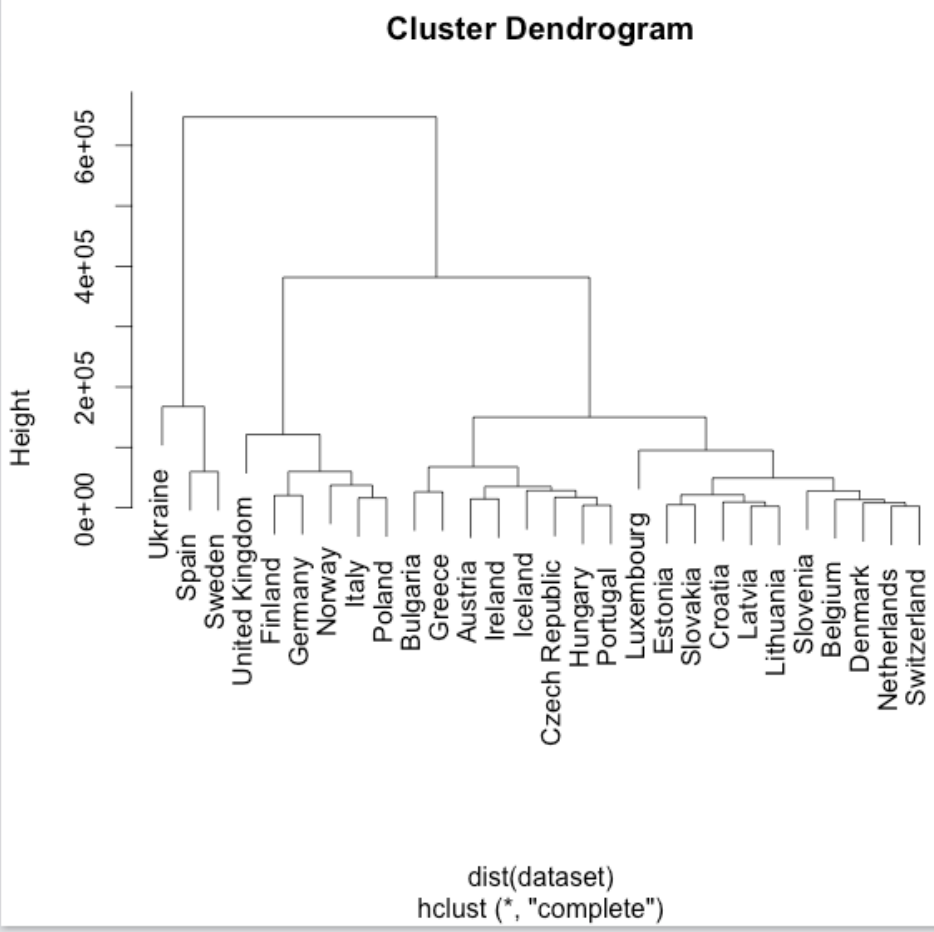
Console ~/Desktop/

```
> dataset <- read.csv(file="europe.csv",head=TRUE,sep=",")
> modelname<-hclust(dist(dataset))
Предупреждение:
В dist(dataset) : в результате преобразования созданы NA
> plot(modelname)
> plot(modelname, labels=dataset$Country)
>
```

Environment History Presentation

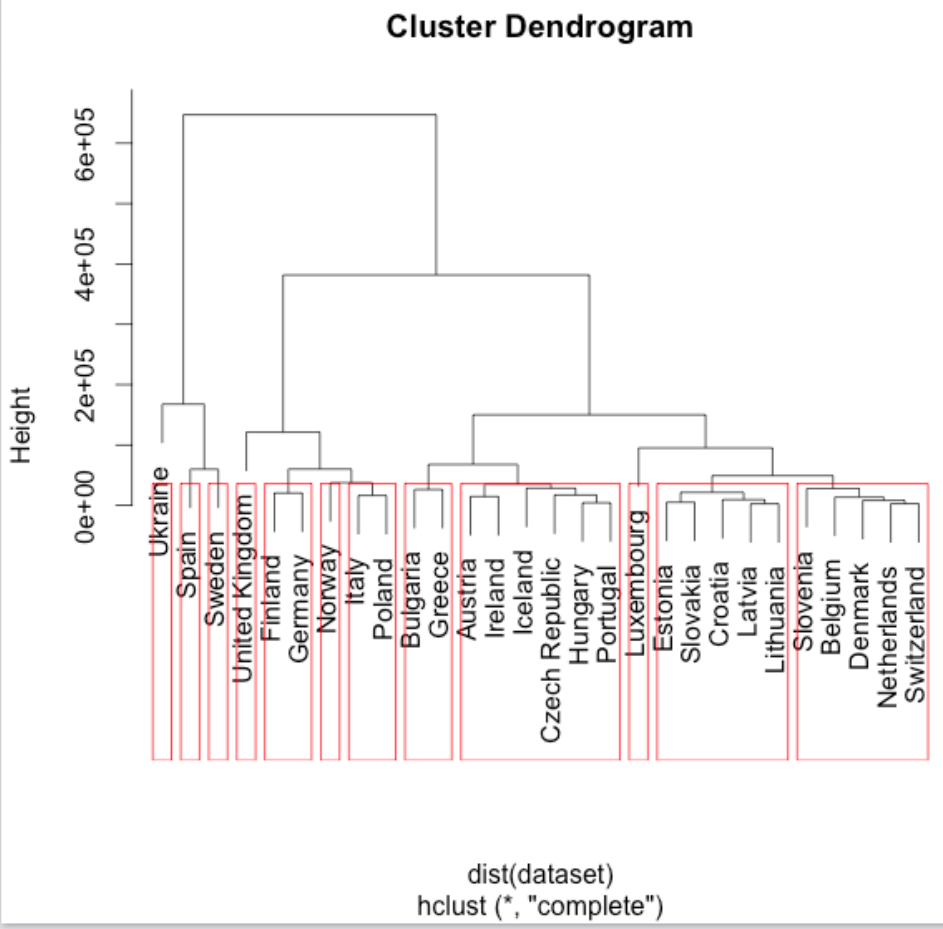
Files Plots Packages Help Viewer

Zoom Export Clear All



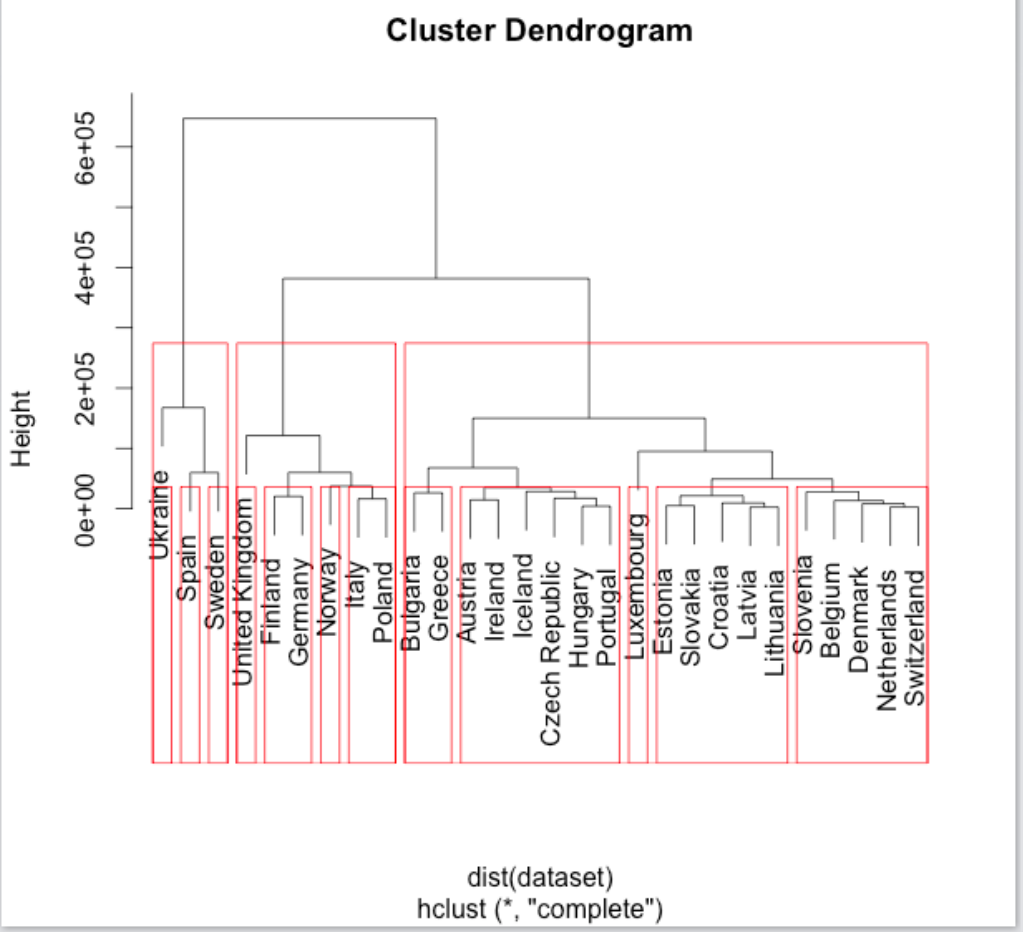
```
dataset <- read.csv(file="europe.csv",head=TRUE,sep=",")
modelname<-hclust(dist(dataset))
plot(modelname)
plot(modelname, labels=dataset$Country)
rect.hclust(modelname, 12)
rect.hclust(modelname, h=2e+05)
```

```
> dataset <- read.csv(file="europe.csv",head=TRUE,sep=",")
> modelname<-hclust(dist(dataset))
Предупреждение:
В dist(dataset) : в результате преобразования созданы NA
> plot(modelname)
> plot(modelname, labels=dataset$Country)
> rect.hclust(modelname, 12)
>
```



```
dataset <- read.csv(file="europe.csv",head=TRUE,sep=",")
modelname<-hclust(dist(dataset))
plot(modelname)
plot(modelname, labels=dataset$Country)
rect.hclust(modelname, 12)
rect.hclust(modelname, h=2e+05)
```

```
> dataset <- read.csv(file="europe.csv",head=TRUE,sep=",")
> modelname<-hclust(dist(dataset))
Предупреждение:
В dist(dataset) : в результате преобразования созданы NA
> plot(modelname)
> plot(modelname, labels=dataset$Country)
> rect.hclust(modelname, 12)
> rect.hclust(modelname, h=2e+05)
>
```



Резюме

- Кластеризация – это автоматическое разбиение множества объектов на группы по принципу схожести
- Общая схема кластеризации одна: выделение характеристик -> выбор метрики -> группировка объектов -> представление результатов. Но существует много различных реализаций этой схемы.
- Кластеризация данных широко используемая технология обработки информации.