Министерство образования и науки Российской Федерации

ФГАОУ ВО «Севастопольский государственный университет»

Институт информационных технологий

и управления в технических системах

Методические указания к выполнению лабораторных и расчетно-графической работ

по дисциплине «Интеллектуальный анализ данных»

для студентов всех форм обучения направления подготовки

09.03.02 «Информационные системы и технологии»

Севастополь

2018

УДК 004.9:528(075.8)

Методические указания к лабораторным и расчетно-графической работм по дисциплине «Интеллектуальный анализ данных» / Сост.: И.П. Шумейко, О.А. Сырых, И.В. Дымченко – Севастополь: Изд-во СевГУ, 2018 – 96 с.

Методические указания предназначены для проведения лабораторных работ по дисциплине «Интеллектуальный анализ данных». Целью методических указаний является помощь студентам в изучении возможностей системы RStudio (R Commander) и платформы Deductor для решения задач интеллектуального анализа данных. Излагаются практические сведения необходимые для выполнения расчетно-графической работы, требования к содержанию отчета и критерии оценивания.

Методические указания рассмотрены и утверждены на методическом семинаре и заседании кафедры «Информационные системы» (протокол № ___ от «___» _____ 20__ г.)

Содержание

Введение 4
Лабораторная работа №1 Исследование возможностей языка R для статистического анализа данных 5
Лабораторная работа №2.1 Корреляционный и регрессионный анализ данных. Создание набора данных. 15
Лабораторная работа №2.2 Корреляционный и регрессионный анализ данных. Работа с диаграммами 27
Лабораторная работа №2.3 Корреляционный и регрессионный анализ данных. Исследование тесноты взаимосвязей данных в среде R 38
Лабораторная работа №2.4 Корреляционный и регрессионный анализ данных. Множественная линейная регрессия. 46
Лабораторная работа №3 Задача дисперсионного анализа. Методы дисперсионного анализа. Однофакторный дисперсионный анализ. 50
Лабораторная работа №4 Кластерный анализ. Основные этапы и задачи кластерного анализа данных. 64
Лабораторная работа №5_1 Линейный дискриминантный анализ. Построение канонических и классификационных функций. 71
Лабораторная работа №5_2 Линейный дискриминантный анализ. Проведение дискриминантного анализа и интерпретация результатов. 74
Задание и методические указания к расчетно-графической работе 76
Требования к содержанию и оформлению отчетов 89
Организация защиты и критерии оценивания выполнения лабораторных работ 90
Список литературы 92
Приложение 1 Образец оформления и содержания отчета по лабораторной работе 93
Приложение 2 Образец единого титульного листа к отчетам по лабораторным работам 94
Приложение 3 Образец титульного листа к отчету по расчетно-графической работе 95

Введение

Целью дисциплины «Интеллектуальный анализ данных» является обучение основам интеллектуального анализа данных для формирования главных понятий и навыков, необходимых при решении сложных междисциплинарных научно-технических, социально-экономических и других задач на базе современных программных средств.

Задачи дисциплины – формирование умений и навыков, позволяющих обучающимся грамотно применять:

- методы первичной обработки и графического представления данных при проведении междисциплинарных исследований;
- информационные технологии структурирования, интерпретации и анализа результатов экспериментов при решении научно-технических, социально-экономических и других задач;
- информационные технологии поиска шаблонов данных;
- инструменты Data Mining и современные пакеты прикладных программ для решения задач обработки экспериментальных данных.

Термин «Анализ данных», или «Интеллектуальный анализ данных» - перевод английского термина «Data Mining», т.е. буквально «добыча данных» или даже «раскапывание данных». То есть исследование и обнаружение "машиной" (алгоритмами, средствами искусственного интеллекта) в сырых данных скрытых знаний, которые ранее не были известны, нетривиальны, практически полезны, доступны для интерпретации человеком.

В настоящее время выделяют следующие основные классы задач анализа данных:

– прогнозирование (Forecasting)
– классификация (Classification)
– кластеризация (Clustering)
– ассоциации (Associations)
– визуализация (Data Visualization)
– обобщение (Summarization): обнаружение отклонений; оценка; анализ/поиск связей

Исследованию методов и инструментов решения таких задач и посвящена тематика лабораторных и расчетно-графической работ по дисциплине «Интеллектуальный анализ данных» при подготовке бакалавров направления 09.03.02 «Информационные системы и технологии».

Лабораторная работа №1
Исследование возможностей языка R для статистического анализа данных

Цель:

- изучить основные особенности языка R;
- исследовать возможности языка R для работы с графикой.

Время: 2 часа

Лабораторное оборудование: персональные компьютеры, выход в сеть Internet, RStudio.

Краткие теоретические сведения

1. Язык R, как инструмент статистического анализа данных

R – статистическая система анализа, созданная Россом Ихакой и Робертом Гентлеманом (1996, J.Comput. Граф. Stat., 5: 299-314). R является и языком и программным обеспечением; его наиболее замечательный особенности:

- эффективная обработка данных и простые средства для сохранения результатов;
- набор операторов для обработки массивов, матриц, и других сложных конструкций;
- большая, последовательная, интегрированная коллекция инструментальных средств для проведения статистического анализа,
- многочисленные графические средства;
- простой и эффективный язык программирования, который включает много возможностей.

Язык R – рассматривают как диалект языка S созданный AT&T Бэлл Лаборатории. S доступен как программное обеспечение S-PLUS коммерческой системы MathSoft (см.http://www.splus.mathsoft.com для получения дополнительной информации). Есть существенные различия в концепции R и S (те, кто хочет знать больше об этом может читать статью, написанную Gentleman и Ihaka (1996) или R-FAQ (часто задаваемые вопросы) (http://cran.r-project.org/doc/FAQ/R-FAQ.html).

R доступен в нескольких формах: исходный текст программ, написанный на C (и некоторые подпрограммы в Fortran77) и в откомпилированном виде.

R – язык со многими функциями для выполнения статистического анализа и графического отображения результатов, которые визуализируется сразу же в собственном окне и могут быть сохранены в различных форматах (например, jpg, png, bmp, eps, или wmf под Windows, ps, bmp, pictex под Unix). Результаты статистического анализа могут быть отображены на экране.

Некоторые промежуточные результаты (P-values, коэффициент регрессии и т.п.) могут быть сохранены в файле и использоваться для последующего анализа.

R – язык, позволяющий пользователю использовать операторы циклов, чтобы последовательно анализировать несколько наборов данных. Также возможно объединить в отдельную программу различные статистические функции, для проведения более сложного анализа.

В библиотеках среды статистического программирования R присутствуют процедуры, реализующие все необходимые методы предварительного анализа и обработки данных. R держит все свои вычисления в оперативной памяти, поэтому если в процессе работы выключится питание, то результаты сессии, не записанные явным образом в файл, пропадут. Эта особенность, к сожалению, также не позволяет R работать с действительно большими объёмами (порядка сотен тысяч и более записей) данных.

2. Графические интерфейсы языка R

Для удобства работы с R разработан ряд графических интерфейсов, в том числе RStudio, JGR, RKWard, SciViews-R, Statistical Lab, R Commander, Rattle.

Кроме того, в ряде текстовых и кодовых редакторов предусмотренные специальные режимы для работы с R, в частности в ConTEXT, Emacs (Emacs Speaks Statistics), jEdit, Kate, Syn, TextMate, Tinn-R, Vim, Bluefish, WinEdt (с пакетом RWinEdt).

Для среды разработки Eclipse существует специализированный R-плагин; доступ к функциям и среде выполнения R возможен из Python с использованием пакета RPy; работать с R можно из эконометрического пакета Gretl.

После установки, для запуска интерпретатора R достаточно выполнить в терминале команду:

```
$ R
```

После чего в консоли появится:

```
R version 3.2.4 (2016-03-10) - "Very Secure Dishes"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)
```

R – это свободное ПО, и оно поставляется безо всяких гарантий. Его можно распространять при соблюдении некоторых условий. Команда 'license()' служит для получения более подробной информации.

R – это проект, в котором сотрудничает множество разработчиков. Команда `'contributors()'` служит для получения дополнительной информации и `'citation()'` для ознакомления с правилами упоминания R и его пакетов в публикациях.

Команда `'demo()'` для запуска демонстрационных программ, `'help()'` – для получения справки, `'help.start()'` – для доступа к справке через браузер.

Для выхода необходимо использовать команду

```
> q()
```

Как видно из выше приведенного листинга для получения подробной информации (справки) о любой функции, необходимо выполнить команду:

```
> help(<имя функции>)
```

либо

```
> ?<имя функции>
```

Например, выполнив одну из команд

```
> help(q)
> ?q
```

получится следующий результат:

```
Terminate an R Session

Description:
```

```
     The function ‘quit’ or its alias ‘q’ terminate the current R
     session.

Usage:

     quit(save = "default", status = 0, runLast = TRUE)
        q(save = "default", status = 0, runLast = TRUE)

Arguments:

    save: a character string indicating whether the environment
          (workspace) should be saved, one of ‘"no"’, ‘"yes"’,
‘"ask"’
          or ‘"default"’.

  status: the (numerical) error status to be returned to the
operating
          system, where relevant.  Conventionally ‘0’ indicates
          successful completion.
 runLast: should ‘.Last()’ be executed?

Details:

     ‘save’ must be one of ‘"no"’, ‘"yes"’, ‘"ask"’ or ‘"default"’.
In
     the first case the workspace is not saved, in the second it is
     saved and in the third the user is prompted and can also decide
     _not_ to quit.  The default is to ask in interactive use but
may
     be overridden by command-line arguments (which must be supplied
in
     non-interactive use)...
```

Кроме функции `help()`, полезной, если неизвестно точное названия функции, может оказаться команда:

```
> help.search("vector")
```

результатом которой, будет список команд, свойственных «векторам», с кратким описанием.

Команда `apropos()` выдаст просто список команд, содержащих строку, которая была в кавычках.

Одним из важных преимуществ R является наличие для него многочисленных расширений (пакетов) практически для решения любой задачи обработки данных, которые можно легко скачать с официального онлайн-репозитория – CRAN (http://cran.r-project.org/) и установить с помощью команды:

```
install.packages()
```

Под Windows есть соответствующий пункт «Установить пакет(ы)» в меню «Пакеты».

Как только пакет установлен, то он сразу готов к работе. Нужно только инициализировать его перед употреблением. Для этого служит команда `library()`.

Под Windows необходимо выполнить команду «Пакеты» → «Включить пакет»

При установке R автоматически устанавливаются так называемые базовые пакеты, без которых система просто не работает (например, это такие пакеты, как `base`, `grDevices`), и некоторые «рекомендованные» пакеты (например, `cluster` (для решения задач кластерного анализа), `nlme` (для анализа нелинейных моделей) и д.р.).

3. Основные особенности языка R

R – регистрозависимый язык, т. е., например, символы «A» и «a» могут обозначать разные объекты.

Для присваивания используется символ «<–» или «–>» (можно также использовать традиционное « =»).

```
2 + 3 -> x -> y          # x = 5; y = 5
z <- x + y               # z = 10
z = z * z           # z = 100
z <- x + y -> t          # z = x+y и t = x+y
```

Аргументы функций передаются в круглых скобках через запятую.

```
func(x, y)          # вызов функции с двумя аргументами x и y
f(g(x, y), y)  # Суперпозиция функций f и g
```

В самом простом случае R можно использовать как «продвинутый» калькулятор:

```
> # Использование R как калькулятора
> 1 + 2 + 3
[1] 6
> exp(1)^exp(1)# e в степени e
[1] 15.15426
> sin(pi/2)
[1] 1
```

Необходимо обратить внимание на единицу в квадратных скобках ([1]) – это индекс элемента вектора. Дело в том, что в R любой результат с числами трактуется как вектор единичной длины, так как скаляров в R, вообще говоря, нет. Кроме этого, нумерация элементов векторов начинается с 1, а не с 0, как во многих других языках программирования.

Порядок арифметических действий в R стандартный, знакомый со школьной математики. Скобки (раскрывающиеся изнутри наружу) позволяют этот порядок действий менять:

Вычислить:

```
> 3/7
> 3/7-0.4285714
> sqrt(2)*sqrt(2)
> (sqrt(2)*sqrt(2))-2
```

И ещё немного о работе с аргументами на примере команды `round()` (округлить). Она имеет два аргумента: число, которое нужно округлить, и значение `digits`, сообщающее, до какого знака округлять. Система аргументов работает разумно, так что все равно, что написать:

```
> round(pi)# Использовали значение по умолчанию для “digits”
[1] 3

> round(pi, 3) # Прямой порядок аргументов
[1] 3.142
```

```
> round(pi, digits = 10)# с использованием имени аргумента
[1] 3.141593

> round(pi, d = 5)# с использованием сокращенного имени
[1] 3.14159

> round(digits = 5, pi)# вызов с другим порядком аргументов
[1] 3.14159
```

Некоторые аргументы могут иметь имена, благодаря чему их можно перечислять не по порядку, а по имени. Имена можно сокращать вплоть до одной буквы, но только если нет других аргументов, которые от такого сокращения станут неразличимы. При перечислении аргументов по порядку имена можно опускать:

Для однострочных комментариев используется символ #:

```
# Комментарий
```

Команды разделяются точкой с запятой «;» или символом перевода на новую строку:

```
1:10 -> a; mean(a);
```

или

```
1:10 -> a
mean(a)
```

Сохранение числовых и строковых значений:

```
> number <- 10 # сохранение объект
> number # выводим объект
[1] 10
> (number <- 10) # Сохраняем и выводим объект
[1] 10
> string <- "Hello" # Сохраняем объект-строку
> string
[1] "Hello"
```

Кроме строк и чисел можно также создавать и сохранять векторы. Вектор создается с помощью функции `c()`, которая объединяет несколько однотипных элементов. Также, с помощью двоеточия «:» или функции `seq()` можно создать регулярную последовательность. Функция `rep()` позволяет повторять некоторый образец.

```
> v1 <- c(2, 3, 4, 6, 10)
> v1
[1]  2  3  4  6 10
> v1[3]  # Получить третий элемент вектора
[1] 4
> v1[3:5] # Получить третий, четвертый и пятый элементы вектора
[1]  4  6 10

> v2 <- c(1:5) # Определить вектор как последовательность от 1 до 5
> v2
[1] 1 2 3 4 5
```

```
> s <- rep("a", 4)
> s
[1] "a" "a" "a" "a"

> rep(1:4, 2)
[1] 1 2 3 4 1 2 3 4
> rep(1:4, each = 2)
[1] 1 1 2 2 3 3 4 4
> (v1 + v2) * v2 # Можно проводить простые операции над векторами
[1]  3 10 21 40 75
> crossprod(v1, v2) # Вычисление скалярного произведения
     [,1]
[1,]   94
> v1[v1 > 4]# Получить все координаты вектора большие 4
[1]  6 10
```

Можно получать различные свойства векторов:

```
> length(v1) # Длина вектора
[1] 5
> mean(v1) # Среднее значение элементов вектора
[1] 5
> var(v1) # Дисперсия элементов вектора
[1] 10
```

Матрицы создаются с помощью команды `matrix()`:

```
> args(matrix)
function (data = NA, nrow = 1, ncol = 1, byrow = FALSE, dimnames =
NULL)
NULL
> matrix(data = 1:5, nrow = 5, ncol = 5, byrow = FALSE) # матрица
заполняется столбцами
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    1    1    1    1
[2,]    2    2    2    2    2
[3,]    3    3    3    3    3
[4,]    4    4    4    4    4
[5,]    5    5    5    5    5
> matrix(data = 1:5, nrow = 5, ncol = 5, byrow = TRUE) # матрица
заполняется строками
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    2    3    4    5
[2,]    1    2    3    4    5
[3,]    1    2    3    4    5
[4,]    1    2    3    4    5
[5,]    1    2    3    4    5
```

Можно создать матрицу с неопределенными значениями (для этого используется специальное обозначение NA):

```
> matrix(data = NA, nrow = 3, ncol = 3)
     [,1] [,2] [,3]
[1,]   NA   NA   NA
```

```
[2,]   NA   NA   NA
[3,]   NA   NA   NA
```

Матрицу можно также получить с помощью функций-комбинаторов `cbind` (объединяет столбцы) и `rbind`(объединяет строки).

```
> m <- cbind(v1, v2) # Создаем матрицу
> m
     v1 v2
[1,]  2  1
[2,]  3  2
[3,]  4  3
[4,]  6  4
[5,] 10  5

> typeof(m) # Получаем тип элементов матрицы
[1] "double"
> class(m) # Получаем класс объекта
[1] "matrix"
> is.matrix(m) # Проверяем, является ли m матрицей
[1] TRUE
> is.vector(m) # m не вектор
[1] FALSE
> dim(m) # Получаем размерность m
[1] 5 2
```

4. Работа с графикой в R

Основной функцией для рисования объектов в R является функция plot(x, y, …):

x – координаты точек графика, либо некоторая графическая структура, функция или объект, содержащий методы рисования.

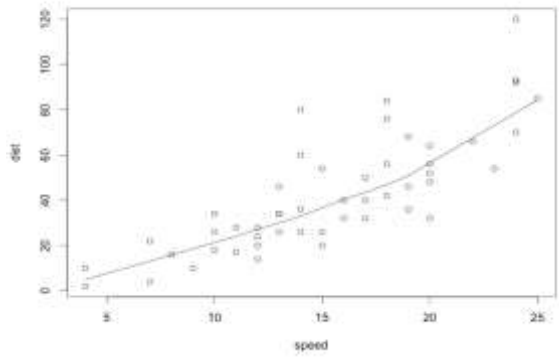
y – y-координаты точек графика, если x – соответствующего типа.

– остальные графические параметры. Перечислим некоторые из них:

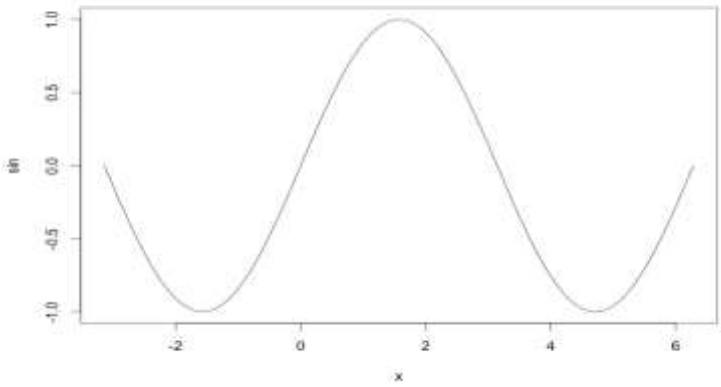
- параметр `type`позволяет изменять внешний вид точек на графике и может принимать одно из следующих значений:
  - `"p"` – точки (*points*; используется по умолчанию);
  - `"l"` – линии (*lines*);
  - `"b"` – изображаются и точки, и линии (*both points and lines*);
  - `"o"` – точки изображаются поверх линий (*points over lines*);
  - `"h"` – гистограмма (*histogram*);
  - `"s"` – ступенчатая кривая (*steps*);
  - `"n"` – данные не отображаются (*no points*).
- параметры `xlab`и `ylab`задают название осей абсцисс и ординат, соответственно;
- параметр `main`задаёт заголовок графика.

Примеры.

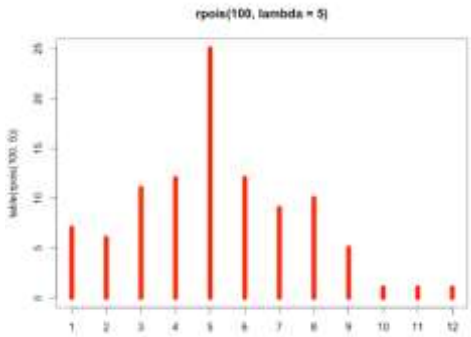
```
>require(stats) # for lowess, rpois, rnorm
plot(cars)
lines(lowess(cars))
```



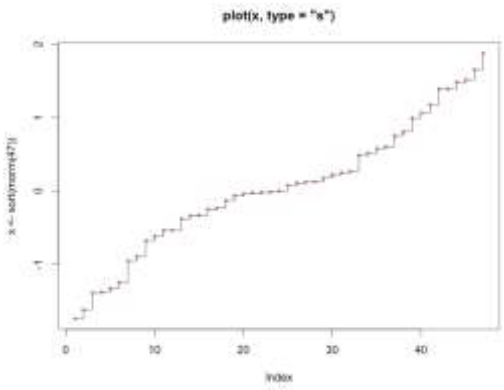

```
>plot(sin, -pi, 2*pi)
```




```
## Discrete Distribution Plot:
> plot(table(rpois(100, 5)), type = "h", col = "red", lwd = 10,
       main = "rpois(100, lambda = 5)")
```




```
## Simple quantiles/ECDF, see ecdf() {library(stats)} for a better
## one:
>plot(x <- sort(rnorm(47)), type = "s", main = "plot(x, type =
\"s\")")
points(x, cex = .5, col = "dark red")
```



Как видно из примеров, с помощью функции plot() можно создавать большое количество разнообразных графиков.

Задание и порядок выполнения лабораторной работы №1

1. Установить R на ПК – https://cran.rstudio.com
2. Установить RStudio – инсталлятор скачать с официального сайта проекта - https://www.rstudio.com/products/rstudio/download3/
3. Ознакомиться с кратким руководством пользователя RStudio – http://r-analytics.blogspot.ch/p/rstudio.html#.WAPrteuvUbf/
4. Исследовать команду `'demo()'`, полученные результаты вставить в отчет
5. Исследовать основные функции и команды языка R, представленные в данной лабораторной работе, полученные результаты вставить в отчет.
6. Ответить на контрольные вопросы

Контрольные вопросы

1. Особенности языка R.
2. Команда для получения подробной информации о функции в R.
3. Структура и особенности команды `round()` в R.
4. Команды для работы с векторами в R (изучить команды, не представленные в методических указаниях).
5. Команды для работы с матрицами в R (изучить команды, не представленные в методических указаниях).
6. Работа с графикой в R (изучить команды, не представленные в методических указаниях).

Лабораторная работа №2.1
Корреляционный и регрессионный анализ данных. Создание набора данных.

Цель:

- исследовать возможности языка R для проведения корреляционного и регрессионного анализа данных;
- создание набора данных для проведения корреляционного и регрессионного анализа данных

Время: 2 часа

Лабораторное оборудование: персональные компьютеры, выход в сеть Internet, RStudio.

Краткие теоретические сведения

Различают два типа связей между различными явлениями и их признаками: функциональную или жестко детерминированную, с одной стороны, и статистическую или стохастически детерминированную - с другой. Строго определить различие этих типов связи можно тогда, когда они получают математическую формулировку.

Важнейшим частным случаем статистической связи является корреляционная связь.

При функциональной связи заданному значению фактора X соответствует строго определенные значение параметра Y, что свойственно строго детерминированным процессам (связь температуры и объема, давления и объема).

При корреляционной связи заданному значению фактора X может соответствовать множество возможных значений параметра Y.

Для изучения взаимосвязей используются корреляционный и регрессионный анализ.

5. Корреляционный анализ

Основной задачей корреляционного анализа является определение формы, направленности и тесноты взаимосвязи. При исследования корреляции используются графический и аналитический подходы.

Графический анализ начинается с построения корреляционного поля. Корреляционное поле (или диаграмма рассеяния) является графической зависимостью между результатами измерений двух признаков. Для ее построения исходные данные наносят на график, отображая каждую пару значений (xi,yi) в виде точки с координатами xi и yi в прямоугольной системе координат.

Визуальный анализ корреляционного поля позволяет сделать предположение о форме взаимосвязи двух исследуемых показателей. По **форме взаимосвязи** корреляционные зависимости принято разделять на линейные (рис. 1) и нелинейные (рис. 2).

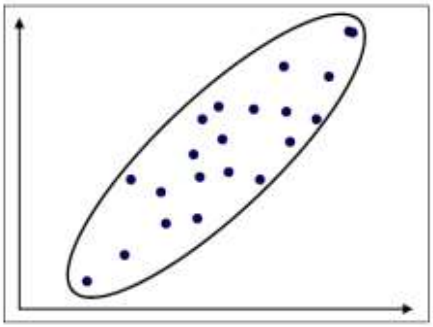


Рис 1. Линейная статистическая связь

При линейной зависимости огибающая корреляционного поля близка к эллипсу. Линейная взаимосвязь двух случайных величин состоит в том, что при увеличении одной случайной величины другая случайная величина имеет тенденцию возрастать (или убывать) по линейному закону.

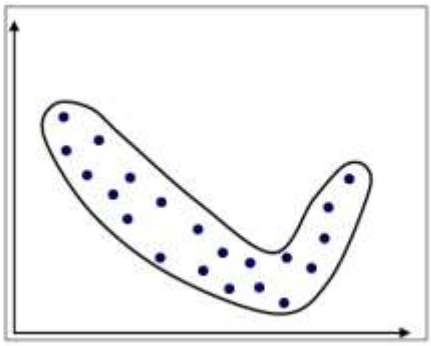


Рис 2. Нелинейная статистическая связь

Выявление формы статистической зависимости необходимо для выбора метода оценки тесноты (силы) взаимосвязи.

Направленность является положительной, если увеличение значения одного признака приводит к увеличению значения второго (рис. 3).

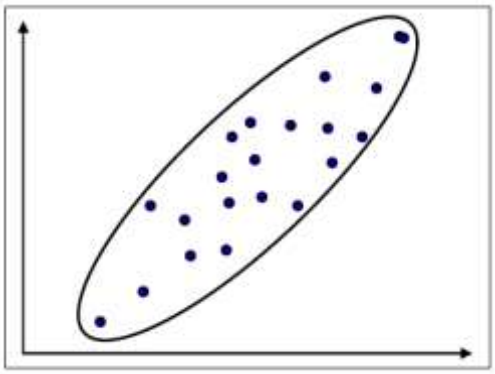


Рис 3. Положительная направленность

Направленность является отрицательной, если увеличение значения одного признака приводит к уменьшению значения второго (рис. 4).

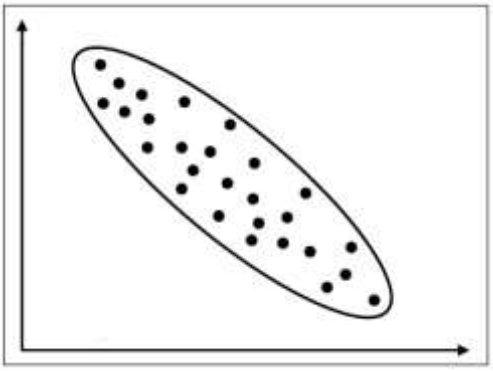


Рис 4. Отрицательная направленность

Теснота взаимосвязи может быть оценена качественно по ширине корреляционного поля – чем меньше его ширина, тем больше теснота и сильнее зависимость.

Количественная оценка тесноты взаимосвязи двух случайных величин осуществляется с помощью коэффициента корреляции *r*.

Коэффициент корреляции характеризует только линейную взаимосвязь

Направление (прямая или обратная) и сила (теснота) корреляционной связи характеризуется **коэффициентом линейной корреляции Пирсона** который рассчитывают по данным выборки *n* объектов по формуле

$$r_{xy} = \frac{n\sum xy - \sum x \sum y}{\sqrt{\left[n\sum x^2 - \left(\sum x\right)^2\right] \cdot \left[n\sum y^2 - \left(\sum y\right)^2\right]}}$$

где *x* - значение факторного признака;

y - значение результативного признака;

n - число пар данных.

Коэффициент корреляции величина относительная; он принимает значение от минус единицы до плюс единицы, т.е. $-1 < r < 1$.

При $r > 0$ связь оценивается, как прямая, при $r < 0$ – обратная.

При $r = 0$ – связь отсутствует, при $|r| = 1$ – связь функциональная

Сила связи оценивается:

при $|r| < 0{,}3$ – как слабая,

при $0{,}3 < [r| < 0{,}7$ – умеренная,

при $|r| > 0{,}7$ – сильная.

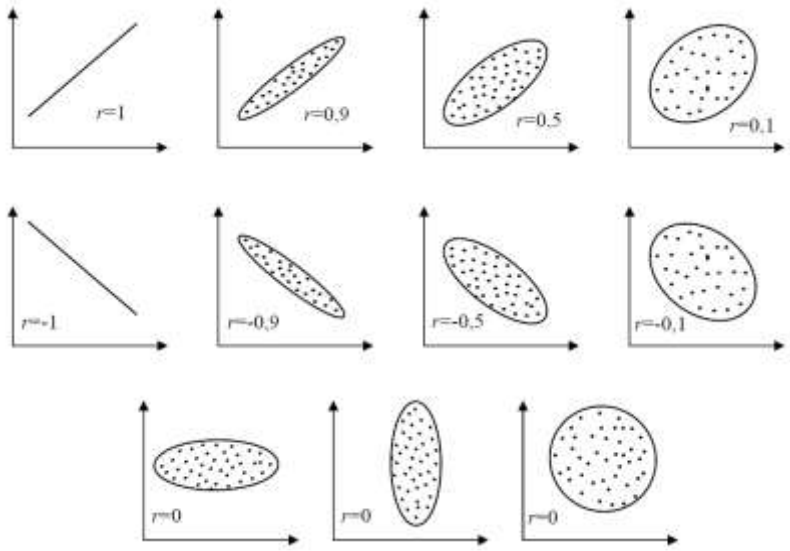


Рис 5. Корреляционные поля при различных значениях коэффициента корреляции.

6. Регрессионный анализ

В практических исследованиях возникает необходимость аппроксимировать (описать приблизительно) диаграмму рассеяния математическим уравнением. То есть зависимость между переменными величинами *Y* и *X* можно выразить аналитически с помощью формул и уравнений и графически в виде геометрического места точек в системе прямоугольных координат.

Основная особенность регрессионного анализа: при его помощи можно получить конкретные сведения о том, какую форму и характер имеет зависимость между исследуемыми переменными.

Под регрессией понимается функциональная зависимость между независимыми (объясняющими) переменными и средним значением зависимой (объясняемой) переменной, которая строится с целью предсказания этого среднего значения при фиксированных значениях переменной.

Этапы регрессионного анализа.

1. Формулировка задачи. На этом этапе формируются предварительные гипотезы о зависимости исследуемых явлений.

2. Определение зависимых и независимых (объясняющих) переменных.

3. Сбор статистических данных. Данные должны быть собраны для каждой из переменных, включенных в регрессионную модель (гипотеза).

4. Формулировка гипотезы о форме связи (простая или множественная, линейная или нелинейная).

5. Определение функции регрессии (заключается в расчете численных значений параметров уравнения регрессии)

6. Оценка точности регрессионного анализа.

7. Интерпретация полученных результатов. Полученные результаты регрессионного анализа сравниваются с предварительными гипотезами. Оценивается корректность и правдоподобие полученных результатов.

8. Предсказание неизвестных значений зависимой переменной.

При помощи регрессионного анализа возможно решение задачи прогнозирования и классификации.

Характер и форма зависимости между переменными могут образовывать следующие разновидности регрессии:

– положительная линейная регрессия (выражается в равномерном росте функции);
– положительная равноускоренно возрастающая регрессия;
– положительная равнозамедленно возрастающая регрессия;
– отрицательная линейная регрессия (выражается в равномерном падении функции);
– отрицательная равноускоренно убывающая регрессия;
– отрицательная равнозамедленно убывающая регрессия.

7. Возможности языка R для проведения корреляционного и регрессионного анализа данных

7.1. Создание набора данных

Первый этап любого анализа данных – создание набора данных, в котором содержится информация для изучения, в подходящем формате. В R эта задача распадается на следующие:

– выбор типа данных;
– ввод или импорт данных в выбранном формате.

Набор данных – это, как правило, прямоугольный массив данных, в котором ряды соответствуют наблюдениям, а столбцы – признакам.

Пример:

Таблица 1. Набор данных о пациентах

Порядковый номер пациента (PatientID)	**Дата поступления: месяц/день/год (AdmDate)**	**Возраст (Age)**	**Тип диабета (Diabetes)**	**Состояние (Status)**
1	10/15/2009	25	Type1	Плохое (Poor)
2	11/01/2009	34	Type2	Улучшившееся (Improved)
3	10/21/2009	28	Type1	Превосходное (Excellent)
4	10/28/2009	52	Type1	Плохое (Poor)

7.1.1. Структура данных

Необходимо различать **структуру** набора данных и **типы** данных, которые его составляют.

R работает с самыми разными структурами данных, включая

– скаляры,

– векторы,
– массивы данных,
– таблицы данных
– списки.

Они различаются типами данных, способом создания, сложностью устройства, а также способом обозначать и извлекать их отдельные элементы. Эти структуры данных схематически изображены на рис. 6

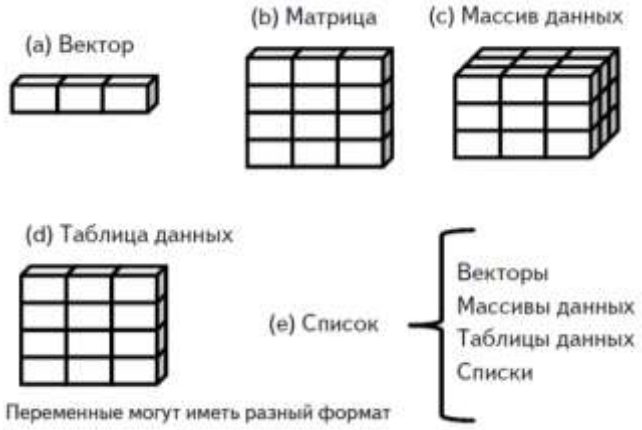


Рис. 6 Типы структуры данных в R

Существует несколько присущих только R терминов.

В R **объектом** (object) называется все, что может быть представлено в виде переменных, включая константы, разные типы данных, функции и даже диаграммы. У объектов есть вид (определяет, в каком виде объект хранится в памяти) и класс (который указывает общим функциям как с ним обращаться).

Таблица данных (data frame) – это тип структуры данных в R, аналогичный тому виду, в котором хранятся данные в обычных статистических программах (Столбцы – это переменные, а строки – это наблюдения. В одной таблице данных могут содержаться переменные разных типов (например, числовые и текстовые). Таблицы данных – это основной тип структуры данных.

Факторы – это номинальные или порядковые переменные. В R они хранятся и обрабатываются особым образом.

Таблица 1 будет прочитана в R как таблица данных.

Типы данных в R бывают
– числовыми (numeric),
– текстовыми (character),
– логическими (TRUE/FALSE, правда/ложь),
– комплексными (мнимое число)
– необработанными (байты).

Переменные PatientID, AdmDate и Age будут прочитаны R как числовые, а Diabetes и Status – как текстовые.

Таблица данных (data frame) – это более широко используемый по сравнению с матрицей объект, поскольку разные столбцы могут содержать разные типы данных (числовой, текстовый и т. д.). Таблица данных – это самая часто используемая структура данных в R.

Набор данных про пациентов (табл. 1) состоит из числовых и текстовых данных. Эти данные нужно представить в виде таблицы данных, а не матрицы, поскольку здесь есть данные разных типов.

Таблица данных создается при помощи функции `data.frame()`:

```
mydata <- data.frame(col1, col2, col3,…),
```

где – col1, col2, col3,… это векторы любого типа (текстового, числового или логического), которые станут столбцами таблицы.

Названия каждому столбцу можно присвоить при помощи функции names().

Пример:

Создание таблицы данных (табл 1)

```
> patientID <- c(1, 2, 3, 4)
> age <- c(25, 34, 28, 52)
> diabetes <- c(“Type1”, “Type2”, “Type1”, “Type1”)
> status <- c(“Poor”, “Improved”, “Excellent”, “Poor”)
> patientdata <- data.frame(patientID, age, diabetes, status)
> patientdata
```

Каждый столбец должен содержать данные только одного типа, при этом в одной таблице данных могут быть столбцы с данными разного типа.

Существует несколько способов обозначить элементы таблицы данных. Можно использовать индексы или можно указывать номера столбцов.

Пример:

Обозначение элементов таблицы данных

1)

```
> patientdata [1:2]
  patientID age
1         1  25
2         2  34
3         3  28
4         4  52
```

2)

```
> patientdata [c("diabetes", "status")]
  diabetes    status
1    Type1      Poor
2    Type2  Improved
3    Type1 Excellent
4    Type1      Poor
```

Знак $ используется, чтобы обозначить определенную переменную в таблице данных.

```
> patientdata$age
[1] 25 34 28 52
```

R названия строк могут быть назначены при помощи параметра row.names функции создания таблицы данных. Например, программный код

patientdata <- data.frame(patientID, age, diabetes, status, row.names=patientID)

назначает patientID переменной, которая будет использоваться для обозначения строк при выводе данных и создании диаграмм в R.

Факторы

Переменные бывают номинальными, порядковыми или непрерывными.

Номинальные переменные – это категориальные данные, которые невозможно упорядочить. Переменная Diabetes – это пример номинальных данных. Даже если обозначить Type 1 (тип 1) единицей, а Type 2 (тип 2) – двойкой, все равно эти цифры нельзя будет сравнивать в терминах «больше – меньше».

Порядковые данные можно упорядочить, но не оценить количественно. Переменная Status – хороший пример порядковых данных. Понятно, что у больного с плохим (poor)

самочувствием дела идут не так хорошо, как у больного, чье состояние улучшилось (improved), но не ясно, насколько.

Непрерывные переменные могут принимать любое значение в пределах определенного диапазона. Их значения можно упорядочить и понять, насколько одно из них больше другого.

Возраст, выраженный в годах, является непрерывной переменной и может принимать такие значения, как 14.5 или 22.8, а также любые значения между этими двумя.

Категориальные (номинальные и порядковые) данные называются в R **факторами**. Факторы очень важны в R, поскольку они определяют, как данные будут проанализированы и графически представлены.

Функция `factor`() сохраняет категориальные данные в виде вектора из целых чисел в диапазоне от одного до *k* (где *k* – число уникальных значений категориальной переменной) и в виде внутреннего вектора из цепочки символов (исходных значений переменной), соответствующим этим целым числам.

Например есть вектор

```
diabetes <- c("Type1", "Type2", "Type1", "Type1").
```

Команда `diabetes <- factor(diabetes)` преобразует этот вектор в (1, 2, 1, 1) и устанавливает внутреннее соответствие 1=Type1 и 2=Type2 (присвоение числовых значений происходит в алфавитном порядке). Любой анализ, который будете проводиться с вектором `diabetes`, будет воспринимать эту переменную как номинальную и выбирать статистические методы, подходящие для этого типа данных.

При работе с векторами, которые представлены порядковыми данными, для функции `factor`() нужно добавлять параметр `ordered=TRUE`. Примененная к вектору

```
status <- c("Poor", "Improved", "Excellent", "Poor")
```

команда `status <- factor(status, ordered=TRUE)` преобразует этот вектор в вид (3, 2, 1, 3) и установит внутреннее соответствие как 1=Excellent, 2=Improved, 3=Poor. Во время любой обработки этого вектора он будет воспринят как порядковая переменная с применением соответствующих статистических методов.

По умолчанию уровни фактора присваиваются значениям вектора в алфавитном порядке. Э

Для упорядоченных факторов редко подходит алфавитный порядок уровней, предлагающийся по умолчанию.

Установку по умолчанию можно изменить при помощи параметра `levels`.

Например,

```
status <- factor(status, order=TRUE,
levels=c("Poor", "Improved", "Excellent"))
```

присвоит уровни значениям вектора следующим образом: 1=Poor, 2=Improved, 3=Excellent.

Пример: Использование факторов

```
> patientID <- c(1, 2, 3, 4)
> age <- c(25, 34, 28, 52)
> diabetes <- c("Type1", "Type2", "Type1", "Type1")
> status <- c("Poor", "Improved", "Excellent", "Poor")
> diabetes <- factor(diabetes)
> status <- factor(status, order=TRUE)
> patientdata <- data.frame(patientID, age, diabetes, status)
```

Сначала вводятся данные как векторы. Затем указывается, что `diabetes` – это фактор, а `status` – это упорядоченный фактор. Потом данные объединяются в таблицу.

```
> str(patientdata)
'data.frame': 4 obs. of 4 variables:
```

```
$ patientID: num 1 2 3 4
$ age : num 25 34 28 52
$ diabetes : Factor w/ 2 levels "Type1","Type2": 1 2 1 1
$ status : Ord.factor w/ 3 levels "Excellent"<"Improved"<..: 3 2
1 3
> summary(patientdata)
patientID age diabetes status
Min. :1.00 Min. :25.00 Type1:3 Excellent:1
1st Qu.:1.75 1st Qu.:27.25 Type2:1 Improved :1
Median :2.50 Median :31.00 Poor :2
Mean :2.50 Mean :34.75
3rd Qu.:3.25 3rd Qu.:38.50
Max. :4.00 Max. :52.00
```

Функция `str(object)` выводит информацию об объекте (в нашем случае это таблица данных).

Ясно видно, что `diabetes` – это фактор, а `status` – это упорядоченный фактор; также указано, как он закодирован внутри программы.

Обратите внимание, что функция `summary`() обрабатывает переменные по-разному. Для непрерывной переменной age вычислены минимум (minimum, Min.), максимум (maximum, Max.), среднее (Mean) и квартили (first and third quartiles: 1st Qu., 3rd Qu.) (Квартили – это числа, которые делят набор данных на четыре равные части (четверти)), а для категориальных переменных diabetes и status подсчитана частота встречаемости каждого значения.

Списки

Списки – это самый сложный тип данных в R. Фактически список – это упорядоченный набор объектов (компонентов). Список может объединять разные (возможно, не связанные между собой) объекты под одним именем. К примеру, список может представлять собой сочетание векторов, матриц, таблиц данных и даже других списков.

Список можно создать при помощи функции `list`():

```
mylist <- list(объект 1, объект 2, …),
```

где объекты – это любые структуры данных

Объектам в списке можно присваивать имена:

```
mylist <- list(name1= объект 1, name2= объект 2, …).
```

Пример. Создание списка

1) Создание списка:

```
>g <- "My First List"
> h <- c(25, 26, 18, 39)
> j <- matrix(1:10, nrow=5)
> k <- c("one", "two", "three")
> mylist <- list(title=g, ages=h, j, k)
```

2) Вывод списка на экран

```
> mylist
$title
[1] "My First List"
$ages
[1] 25 26 18 39
[[3]]
[,1] [,2]
[1,] 1 6
[2,] 2 7
```

```
[3,] 3 8
[4,] 4 9
[5,] 5 10
[[4]]
[1] "one" "two" "three"
```

3) Вывод на экран второго объекта списка

```
> mylist[[2]]
[1] 25 26 18 39
> mylist[["ages"]]
[[1] 25 26 18 39
```

В данном примере создаете список из четырех компонентов: тестовая строка, числовой вектор, матрица и текстовый вектор. В виде списка можно сохранять любое число объектов.

Можно обозначать элементы списка, указав их номер или название внутри двойных квадратных скобок. В данном случае и `mylist[[2]]` и `mylist[["ages"]]` обозначают один и тот же числовой вектор из четырех элементов.

Списки – это важный тип структуры данных в R по двум причинам. Во-первых, они позволяют без труда упорядочить и вызвать на экран разрозненную информацию. Во-вторых, результаты выполнения многих команд представляют собой списки.

В этом случае пользователь извлекает из таких списков нужную информацию.

7.1.2. Ввод данных

Обычно аналитики сталкиваются с данными, которые поступают из разных источников и в разных форматах. Задача состоит в том, чтобы импортировать данные в программу, проанализировать их и представить отчет о результатах. В R реализованы разные способы импорта данных.

На рис. 7 видно, что в R можно вводить данные с клавиатуры, импортировать из текстовых файлов, из Microsoft Excel и Access, из распространенных статистических программ, специализированных форматов, а также из разных систем управления базами данных.

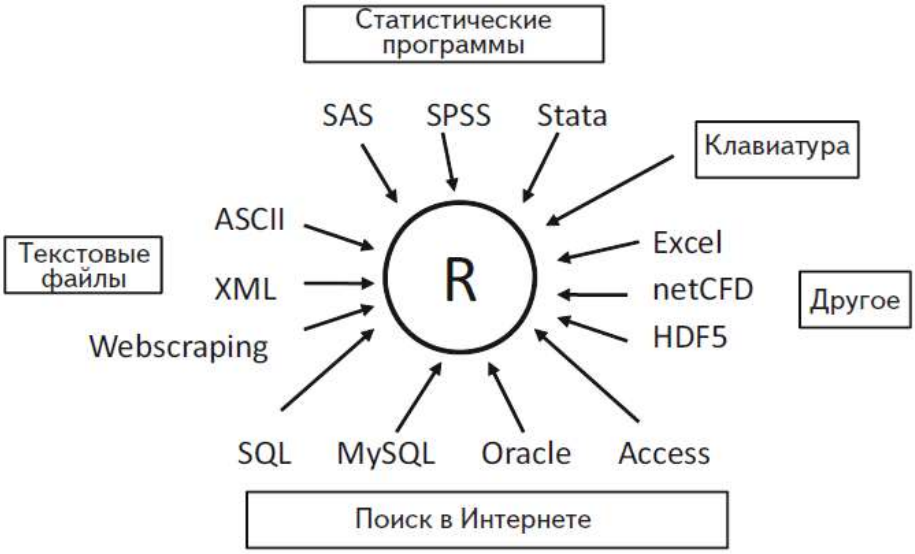


Рис. 7. Источники, из которых можно импортировать данные в R

Ввод данных с клавиатуры

Самый простой способ введения данных – это ввод с клавиатуры. Функция `edit()` откроет текстовый редактор, куда можно внести свои данные.

Для ввода данных необходимо::

1. Создать пустую таблицу данных (или матрицу), указав названия и типы переменных.

2. Открыть текстовый редактор с этим объектом, ввести экспериментальные данные и сохранить результат в виде объекта с данными.

Пример: необходимо создать таблицу данных с названием `mydata` с тремя переменными: `age` (возраст, числовая), `gender` (пол, текстовая) и `weight` (вес, числовая). Затем открыть текстовый редактор, внестие данные и сохранить результат.

```
mydata <- data.frame(age=numeric(0),
gender=character(0), weight=numeric(0))
mydata <- edit(mydata)
```

Функция `edit()` работает с копией объекта. Если не присвоить результат ее работы какому-либо объекту, все изменения пропадут!

Результат работы функции `edit()` под Windows показан на рис. 8.

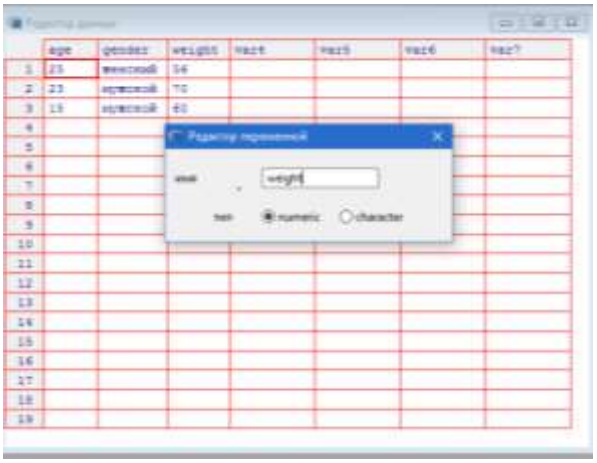


Рис. 8. Редактирование данных при помощи встроенного текстового редактора под Windows

Щелкая по названиям столбцов, можно изменить название и тип соответствующей переменной. Можно добавлять дополнительные переменные, щелкая на названия неиспользованных столбцов.

После закрытия текстового редактора, результаты сохраняются в виде выбранного объекта (в данном случае объект mydata).

Повторное введение функции `mydata <- edit(mydata)` позволяет редактировать введенные данные и добавлять новые.

Импорт данных из текстового файла с разделителями

Импорт данных из текстовых файлов с разделителями возможен при помощи команды `read.table()`, функции, которая сохраняет данные в виде таблицы.

```
mydataframe <- read.table(file, header=логическое_значение,
sep="разделитель", row.names="название")
```

где file – это ASCII файл с разделителями, header – это логическое значение, определяющее, содержит ли первая строка названия переменных (TRUE – да, FALSE – нет), sep указывает, каким символом разделены элементы данных, а row.names – необязательный параметр, для указания столбца (столбцов), в котором содержатся названия строк.

Пример, программный код

```
grades <- read.table("studentgrades.csv", header=TRUE, sep=",",
row.names="STUDENTID")
```

позволяет прочесть файл с разделителями-запятыми, который называется studentgrades.csv, из текущей рабочей директории и сохранить его в виде таблицы данных с

названием grades. В этом файле названия переменных содержались в первой строке, а названия строк – в столбце с названием STUDENTID.

использование параметра `sep` позволяет импортировать файлы с любыми символами в качестве разделителей.

По умолчанию текстовые переменные преобразуются в факторы. Такое преобразование можно заблокировать разными способами. Добавление параметра `stringsAsFactors=FALSE` не позволит преобразовывать в факторы все текстовые переменные. В качестве альтернативы можно использовать параметр `colClasses` для того, чтобы указать формат (например, логический, числовой, текстовый, фактор) каждого столбца.

У функции `read.table()` есть много дополнительных параметров, при помощи которых можно контролировать импорт данных. Подробнее об этом можно прочесть, выполнив команду `help(read.table)`.

Импорт данных из Excel

Лучший способ прочесть файл в формате Excel – это сохранить его в формате текстового файла с разделителями и импортировать в R, как это описано выше. Под Windows для доступа к файлам Excel также можно использовать пакет RODBC. В первой строке электронной таблицы должны содержаться названия переменных (столбцов).

Прежде всего необходимо скачать и установить пакет RODBC.

```
install.packages("RODBC")
```

Теперь можно использовать следующий программный код для импорта данных:

```
library(RODBC)
channel <- odbcConnectExcel("myfile.xls")
mydataframe <- sqlFetch(channel, "mysheet")
odbcClose(channel)
```

Здесь `myfile.xls` – это файл Excel, `mysheet` – это название нужного листа из рабочей книги Excel, `channel` – это вспомогательный объект RODBC, созданный функцией `odbcConnectExcel()`, и `mydataframe` – это получившаяся таблица данных. Этот пакет можно также использовать для импорта данных из Microsoft Access. Подробности изложены в файле справки: `help(RODBC)`.

В Excel 2007 используются файлы формата XLSX, которые фактически представляют собой сжатый набор XML-файлов. Для импорта электронных таблиц в этом формате можно использовать пакет `xlsx`.

Функция `read.xlsx()` осуществляет импорт нужного листа XLSX-файла в таблицу данных. Проще всего использовать эту функцию по такой схеме: `read.xlsx(file, n)`, где `file` – это путь к файлу книги Excel 2007, а `n` – число листов, которые нужно импортировать. Например, под Windows программный код

```
library(xlsx)
workbook <- "c:/myworkbook.xlsx"
mydataframe <- read.xlsx(workbook, 1)
```

импортирует первый лист книги `myworkbook.xlsx`, хранящейся на диске C:, и сохраняет его в виде таблицы данных `mydataframe`. Пакет `xlsx` может не только импортировать листы XLSX-файлов. Он также может создавать файлы этого формата и управлять ими.

Задание и порядок выполнения лабораторной работы №2.1

7. Ознакомится с методическими указаниями,
8. Исследовать основные функции и команды языка R, представленные в данной лабораторной работе
9. Выполнить все примеры.
10. Подобрать экспериментальные данные для анализа (пример данных представлен в Приложении А)
11. Выполнить ввод данных с клавиатуры,
12. Провести экспорт данных из текстового файла с разделителями,
13. Выполнить экспорт данных из Excel.

Контрольные вопросы

1. Функциональная связь.
2. Статистическая связь.
3. Корреляционная связь.
4. Корреляционный анализ.
5. Корреляционное поле.
6. Корреляционный анализ: форма зависимости.
7. Корреляционный анализ: направленность взаимосвязи.
8. Корреляционный анализ: теснота (сила) взаимосвязи.
9. Коэффициент корреляции. Его свойства.
10. Регрессионный анализ.
11. Этапы регрессионного анализа.

Лабораторная работа №2.2
Корреляционный и регрессионный анализ данных. Работа с диаграммами

Цель:

- исследовать возможности языка R для проведения корреляционного и регрессионного анализа данных;
- исследовать возможности языка R для создания и изменения вида диаграмм

Время: 2 часа

Лабораторное оборудование: персональные компьютеры, выход в сеть Internet, RStudio.

Краткие теоретические сведения

Работа с диаграммами

R – это изумительная программа для построения диаграмм. В стандартной интерактивной сессии создается диаграмма, вводом по одной команде и добавлением элементов диаграммы, пока не получится то, что необходимо.

Пример.

Рассмотрим следующие пять строк:

```
attach(mtcars)
plot(wt, mpg)
abline(lm(mpg~wt))
title("Regression of MPG on Weight")
detach(mtcars)
```

Первая команда добавляет в траекторию поиска таблицу данных `mtcars` (встроенный набор данных). Вторая команда открывает окно графики и создает диаграмму рассеяния, на которой вес автомобиля отложен на горизонтальной оси, а расход топлива – на вертикальной. Третья команда добавляет регрессионную прямую. Четвертая команда добавляет название. Последняя команда удаляет таблицу данных из пути поиска. В R диаграммы обычно создаются в таком интерактивном стиле (см. рис.1).

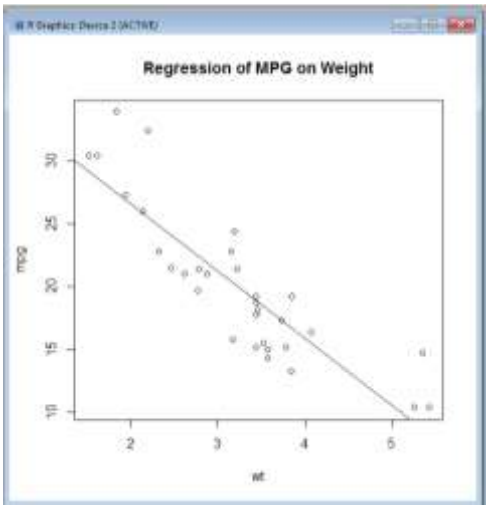


Рисунок 1.

Диаграммы можно сохранять при помощи программного кода или меню графического пользовательского интерфейса. Для сохранения диаграммы при помощи кода необходимо разместить создающие диаграмму команды между командами, которые назначают место вывода и закрывают вывод. Например, следующий программный код позволяет сохранить диаграмму в формате PDF под названием mygraph.pdf в текущей рабочей директории:

```
pdf("mygraph.pdf")
attach(mtcars)
plot(wt, mpg)
abline(lm(mpg~wt))
title("Regression of MPG on Weight")
detach(mtcars)
dev.off()
```

В дополнение к pdf() можно использовать функции win.metafile(), png(), jpeg(), bmp(), tiff(), xfig() и postscript(), чтобы сохранять диаграммы в других форматах. Необходимо учитывать, что формат Windows metafile доступен только под Windows.

Способ сохранения диаграмм при помощи графического пользовательского интерфейса различается в зависимости от оперативной системы. Под Windows при активированном графическом устройстве нужно выбрать в меню Файл → Сохранить как, а затем в появившемся диалоге выбрать нужный формат графического файла и директорию для сохранения.

Новая диаграмма, которая создается при помощи команды высокого уровня, такой как plot(), hist() или boxplot(), обычно заменяет предыдущую диаграмму. Для создания более одной диаграммы есть несколько способов:

1. Открыть новое графическое устройство, перед тем как создавать новую диаграмму

```
dev.new()
команды для построения диаграммы 1
dev.new()
команды для построения диаграммы 2
и т.д.
```

Каждая новая диаграмма будет появляться в последнем открытом окне.

2. Можно получить доступ к нескольким диаграммам сразу через пользовательский интерфейс. Под Windows эта операция состоит из двух этапов. После того как открыто первое окно графики, необходимо выбрать в меню История команд → Запись. Затем использовать пункты меню Предыдущий и Следующий для перемещения между созданными диаграммами.

3. Можно использовать функции dev.new(), dev.next(), dev.prev(), dev.set() и dev.off() для одновременного открытия нескольких окон графики и выбора необходимой диаграммы.

Пример.

В табл. 1. Представлен набор данных, который описывает реакцию пациента на два лекарства в пяти дозировках.

Таблица 1. Реакция пациента на два лекарства в пяти дозировках

Дозировка	Реакция на лекарство A	Реакция на лекарство B
20	16	15
30	20	18
40	27	25
45	40	31
60	60	40

Эти данные можно ввести при помощи следующего программного кода:

```
dose <- c(20, 30, 40, 45, 60)
drugA <- c(16, 20, 27, 40, 60)
drugB <- c(15, 18, 25, 31, 40)
```

Простой линейный график, изображающий зависимость реакции пациента от дозы лекарства A, можно создать так:

```
plot(dose, drugA, type="b")
```

`plot()` – это функция общего назначения, которая строит диаграммы в R. В этом случае `plot(x, y, type="b")` располагает *x* на горизонтальной оси, а *y* – на вертикальной, изображает точки с координатами (*x*, *y*) и соединяет их линиями.

Параметр `type="b"` означает, что на графике должны быть показаны и точки, и линии.

Получившийся график показан на рис. 2.

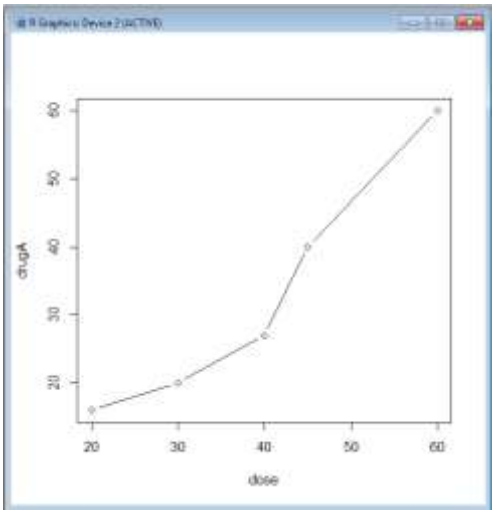


Рисунок 2. График зависимости реакции пациента от дозы лекарства А

Графические параметры

Многие характеристики диаграмм (шрифты, цвета, оси, названия) можно изменять при помощи опций, которые называются «графические параметры».

Один способ назначить эти параметры – использовать функцию `par()`. Значения параметров, заданные таким способом, будут действовать на протяжении всей сессии, пока вы не измените их. Формат применения функции таков: `par(название параметра=назначение, название параметра=назначение, ...)`. Функция `par()` без аргументов выводит на экран действующие значения графических параметров. Добавление аргумента `no.readonly=TRUE` позволяет увидеть только те графические параметры, которые можно изменять.

Например для обозначения отдельных пациентов нужно использовать заполненный треугольник вместо пустого кружка и соединить символы пунктирной линией, а не сплошной. Это можно сделать при помощи следующего программного кода:

```
opar <- par(no.readonly=TRUE)
par(lty=2, pch=17)
plot(dose, drugA, type="b")
par(opar)
```

Получившийся график показан на рис. 3.

Первая команда создает копию текущих параметров. Вторая команда назначает тип линии – пунктирная (lty=2) вместо сплошной по умолчанию и тип символа – заполненный треугольник (pch=17). Затем создается график и восстанавливаются исходные значения параметров.

Можно использовать столько функций `par()`, сколько нужно, так что команда может быть также записана в виде

```
par(lty=2)
par(pch=17)
```

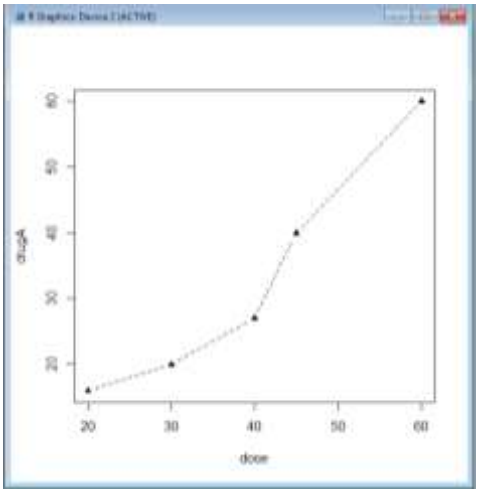


Рисунок 3. График зависимости реакции пациента от дозы лекарства А с измененными типом линии и символами.

Второй способ задать графические параметры – это включить записи типа `название параметра=назначение` внутрь графической функции высокого уровня. В этом случае заданные параметры будут действовать только для конкретной диаграммы. Можно было бы построить тот же график при помощи следующего программного кода:

```
plot(dose, drugA, type="b", lty=2, pch=17)
```

Не во всех графических функциях высокого уровня можно изменять все возможные графические параметры.

Познакомьтесь со справкой по каждой функции для построения диаграмм (например: `?plot`, `?hist` или `?boxplot`), чтобы узнать, какие графические параметры можно назначать таким образом.

Символы и линии

Графические параметры можно использовать для того, чтобы указывать тип символов и линий на диаграммах. Соответствующие параметры перечислены в табл. 2.

Таблица 2. Параметры для указания типов символов и линий

Параметр	Описание
`pch`	Определяет тип символа (см. рис.4)
`cex`	Определяет размер символа. `cex`– это число, обозначающее, как символы должны быть масштабированы по отношению к размеру по умолчанию. 1 = размер по умолчанию, 1.5 – на 50% крупнее, 0.5 – на 50% мельче и т. д.
`lty`	Определяет тип линии (см. рис. 5)
`lwd`	Определяет толщину линии по сравнению с толщиной линии по умолчанию (1). Например, lwd=2 делает линию в два раза толще, чем по умолчанию

Параметр `pch=` определяет тип символов, которые используются на диаграмме. Возможные значения приведены на рис. 4.

Для символов с 21 по 25 можно отдельно указывать цвет контура (`border=`) и заполнения (`bg=`).

Рисунок 4. Символы, назначаемые при помощи параметра `pch`.

Параметр `lty=` применяется для обозначения нужного типа линии. Значения параметра показаны на рис. 5.

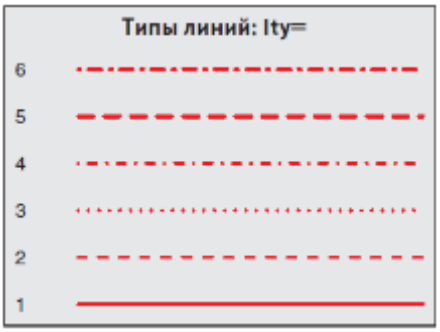


Рисунок 5. Типы линий, назначаемые при помощи параметра `lty`.

Программный код, объединяющий все эти параметры,

```
plot(dose, drugA, type="b", lty=3, lwd=3, pch=15, cex=2)
```

создаст график, на котором точечная линия в три раза шире, чем по умолчанию, соединяет наблюдения, представленные в виде заполненных квадратов в два раза большего размера, чем по умолчанию.

Результат представлен на рис. 6.

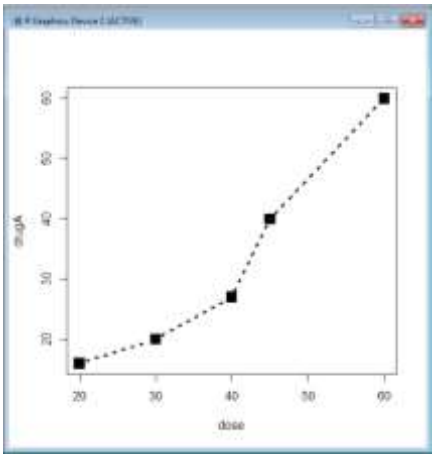


Рисунок 6. График зависимости реакции пациента от дозы лекарства А с измененными типом и шириной линии, а также типом и размером символов.

Цвета

В R есть несколько связанных с цветами параметров. В табл. 3 приведены некоторые из самых распространенных.

Таблица 3. Параметры для назначения цвета

Параметр	**Описание**
col	Цвет элементов на графике. Для некоторых функций (таких как `lines` и `pie`) можно указывать вектор из значений, которые используются по очереди. Например, если `col=c("red", "blue")` и изображены три линии, первая будет красной, вторая – синей и третья – красной.
col.axis	Цвет значений осей
col.lab	Цвет подписей осей
col.main	Цвет заголовков
col.sub	Цвет подзаголовков
fg	Цвет графика
bg	Цвет фона

В R цвета можно обозначать номером, названием, в шестнадцатеричной системе, а также в системах RBG или HSV. Например, `col=1`, `col="white"`, `col="#FFFFFF"`, `col=rgb(1,1,1)` и `col=hsv(0,0,1)` – взаимозаменяемые способы обозначить белый цвет. Функция `rgb`() определяет цвета по значениям красного, зеленого и синего, а `hsv`() основана на значениях оттенка и насыщенности.

Функция `colors`() выводит на экран список всех доступных цветов.

В R также реализован ряд функций, которые позволяют создавать векторы из близких цветов. К таким функциям относятся `rainbow`(), `heat.colors`(), `terrain.colors`(), `topo.colors`() и `cm.colors`(). Например, `rainbow`(10) создает 10 соседних "радужных" цветов. Оттенки серого создаются функцией gray().

В этом случае задаются оттенки серого в виде вектора чисел от 0 до 1. Команда `gray(0:10/10)` создаст 10 оттенков серого.

Пример: чтобы увидеть, как это работает можно запустить программный код

```
n <- 10
mycolors <- rainbow(n)
pie(rep(1, n), labels=mycolors, col=mycolors)
mygrays <- gray(0:n/n)
pie(rep(1, n), labels=mygrays, col=mygrays)
```

Характеристики текста

Графические параметры также используются для определения размера, шрифта и стиля текста. Параметры, определяющие размер шрифта, приведены в табл. 4. Параметры, при помощи которых можно указать тип шрифта, перечислены в табл. 5.

Таблица 4. Параметры, определяющие размер шрифта

Параметр	Описание
`cex`	Число, определяющее, как отображаемый на диаграмме текст будет масштабирован относительно размера по умолчанию (1). 1.5 – на 50% больше, 0.5 – на 50% меньше и т. д.
`cex.axis`	Размер значений на осях по отношению к `cex`
`cax.lab`	Размер названий осей по отношению к `cex`
`cex.main`	Размер заголовков по отношению к `cex`
`cex.sub`	Размер подзаголовков по отношению к `cex`

Например, на всех диаграммах, созданных после команды

```
par(font.lab=3, cex.lab=1.5, font.main=4, cex.main=2)
```

в 1.5 раза более крупные, чем по умолчанию, подписи осей будут выделены курсивом, а названия будут в два раза крупнее, чем по умолчанию, и еще выделены полужирным курсивом.

Таблица 5. Параметры, определяющие семейство, размер и стиль шрифта.

Параметр	Описание
`font`	Число, которое определяет шрифт для текста на диаграмме. 1 = обычный, 2 = полужирный, 3 = курсив, 4 = полужирный курсив, 5 = символы (в кодировке Adobe)
`font.axis`	Шрифт значений на осях
`font.lab`	Шрифт для подписей по осям
`font.main`	Шрифт для заголовков
`font.sub`	Шрифт для подзаголовков
`ps`	Размер точки в шрифте (приблизительно 0.3 мм)
`family`	Семейство шрифтов. Стандартные значения – `serif`, `sans` и `mono`

Размер и стиль шрифта установить просто, тогда как с семейством шрифтов дело обстоит немного сложнее. Это происходит потому, что отображение `serif`, `sans` и `mono` зависит от устройства. Например, под Windows `mono` отображается как TT Courier New, `serif` – как TT Times New Roman, а `sans` – как TT Arial (TT обозначает шрифт типа True Type). Если вы удовлетворены таким отображением семейств шрифтов, то можете использовать параметры типа `family="serif",` чтобы добиться желаемого результата. Если вы не удовлетворены, вам нужно создать новую систему соответствий. Под Windows можете назначать эти соответствия при помощи функции `windowsFont`().

Например, после выполнения команды

```
windowsFonts(
A=windowsFont("Arial Black"),
B=windowsFont("Bookman Old Style"),
C=windowsFont("Comic Sans MS")
)
```

можно использовать A, B и C как названия семейств шрифтов.

В этом случае `par(family="A")` назначит шрифт Arial Black

Размеры диаграммы и полей

Можно определять размер диаграммы и полей при помощи параметров, приведенных в табл. 6.

Таблица 6. Параметры для определения размеров диаграммы и полей

Параметр	**Описание**
`pin`	Размер диаграммы (ширина, высота) в дюймах
`mai`	Числовой вектор, задающий размеры полей, где параметры `c(низ, лево, верх, право)` измеряются в дюймах
`mar`	Числовой вектор, задающий размеры полей, где `c(низ, лево, верх, право)` измеряются в числе строк. По умолчанию это `c(5, 4, 4, 2) + 0.1`

Команда `par(pin=c(4,3), mai=c(1,.5, 1, .2))` позволяет создавать диаграммы размером 4 дюйма в ширину и 3 дюйма в высоту с шириной полей сверху и снизу по одному дюйму, слева 0.5 дюйма и справа 0.2 дюйма.

Пример. Представленный ниже программный код позволяет получить диаграммы, показанные на рис. 7

```
dose <- c(20, 30, 40, 45, 60)
drugA <- c(16, 20, 27, 40, 60)
drugB <- c(15, 18, 25, 31, 40)
opar <- par(no.readonly=TRUE)
par(pin=c(2, 3))
par(lwd=2, cex=1.5)
par(cex.axis=.75, font.axis=3)
plot(dose, drugA, type="b", pch=19, lty=2, col="red")
plot(dose, drugB, type="b", pch=23, lty=6, col="blue",
bg="green")
par(opar)
```

Сначала вводятся данные в виде векторов, затем сохраняются текущие графические параметры (чтобы можно было восстановить их позднее). Затем происходит изменение графических параметров так, чтобы графики были 2 дюйма в ширину и 3 дюйма в высоту. Кроме того, линии будут в два раза шире, а символы – в 1.5 раза крупнее, чем по умолчанию.

Значения на осях даны курсивом, их размер составляет 75% от размера по умолчанию. Затем создан первый график с заполненными красными кружками и пунктирными линиями. Второй график содержит зеленые ромбы с синей каймой и синие линии. И восстанавление исходных графических параметров.

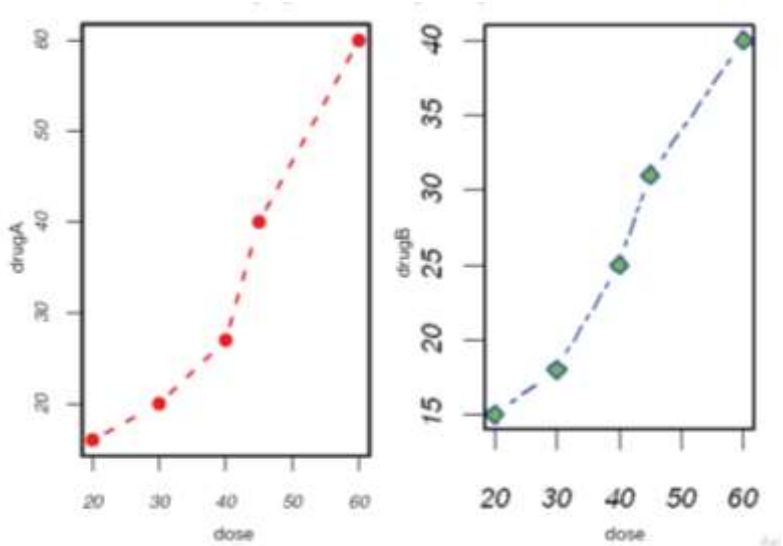


Рисунок 7. Линейный график зависимости реакции пациента от дозы лекарств А и В

Параметры, заданные при помощи функции `par`(), применяются к обоим графикам, а параметры, назначенные ≪внутри≫ функций `plot`(), действуют только для соответствующего графика.

Добавление текста, настройка параметров осей и условных обозначений

Для многих графических функций высокого уровня (например, `plot`, `hist`, `boxplot`) возможен контроль не только графических параметров, но и параметров осей и надписей. К примеру, при помощи приведенного ниже программного кода можно разместить на диаграмме заголовок (`main`), подзаголовок (`sub`) и подписи осей (`xlab`, `ylab`), а также задать диапазон значений на осях (`xlim`, `ylim`). Результат представлен на рис. 8.

```
plot(dose, drugA, type="b",
```

```
col="red", lty=2, pch=2, lwd=2,
main="клинические испытания прпарата А",
sub="это вымышленные данные",
xlab="Доза", ylab="Эффект от препарата",
xlim=c(0, 60), ylim=c(0, 70))
```

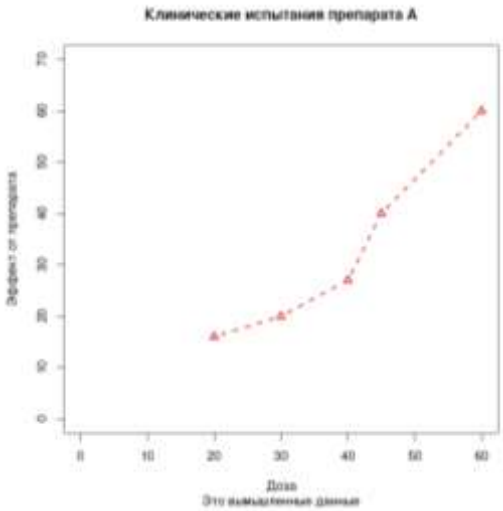


Рисунок 8. Линейный график зависимости реакции пациента от дозы лекарства А с заголовком, подзаголовком и модифицированными осями

Некоторые графические функции высокого уровня по умолчанию выводят надписи и подписи на диаграммах. От них можно избавиться, указав `ann=FALSE` как один из аргументов команд `plot()` или `par()`.

Заголовки

Для размещения заголовков и подписей осей на диаграмме используется функция `title()`. Формат ее применения таков:

```
title(main="Мой_заголовок", sub="подзаголовок",
xlab="подпись_по_оси_x", ylab=" подпись_по_оси _y»)
```

Графические параметры (такие как размер и тип шрифта, ориентация и цвет текста) также можно задать при помощи функции `title()`.

Пример: следующий программный код позволяет получить диаграмму с красным заголовком, синим подзаголовком и зелеными подписями по осям, размер которых на 25% меньше, чем по умолчанию:

```
title(main=" Мой_заголовок " , col.main="red",
sub="мой подзаголовок " , col.sub="blue",
xlab="моя_подпись_по_оси_x ", ylab="моя_подпись_по_оси_Y",
col.lab="green", cex.lab=0.75)
```

Оси

Вместо осей, создаваемых на диаграммах по умолчанию, можно создать оси по своему усмотрению, используя функцию `axis()`. Формат ее применения таков (все параметры описаны в табл. 7):

```
axis(side, at=, labels=, pos=, lty=, col=, las=, tck=, ...).
```

Таблица. 3.7. Параметры осей

Параметр	Описание
`side`	Цифра, определяющая, с какой стороны диаграммы рисовать ось (1 = низ, 2 = лево, 3 = верх, 4 = право)

`at`	Числовой вектор, который задает положение делений на осях
`labels`	Текстовый вектор, который содержит подписи под делениями осей (если вектор не задан, используются значения вектора `at`)
`pos`	Координата оси (то есть значение другой оси, в котором первая ось пересекает ее)
`lty`	Тип линии
`col`	Цвет линии и делений оси
`las`	Положение подписей делений по отношению к оси (0 = параллельно, 2 = перпендикулярно)
`tck`	Длина деления оси, выражается в виде доли от длины диаграммы (отрицательное число означает положение деления кнаружи от рамки диаграммы, положительное число – внутри рамки диаграммы, 0 – отсутствие делений, 1 – сетка); значение по умолчанию –0.01

При создании осей нужно предотвратить появление осей, которые создаются по умолчанию графической функцией высокого уровня. Аргумент `axes=FALSE` подавляет создание всех осей (даже рамки вокруг диаграммы, если не добавлен аргумент `frame.plot=TRUE`). Аргументы `xaxt="n"` и `yaxt="n"` отменяют создание *x*- и *y*-осей соответственно (при этом рамка без делений остается).

Задание и порядок выполнения лабораторной работы №2.2

14. Ознакомится с методическими указаниями,
15. Исследовать основные функции и команды языка R, представленные в данной лабораторной работе
16. Выполнить все примеры.
17. Реализовать код и прописать комментарии к каждому действию

```
x <- c(1:10)
y <- x
z <- 10/x
opar <- par(no.readonly=TRUE)
par(mar=c(5, 4, 4, 8) + 0.1)
plot(x, y, type=”b”,
pch=21, col=”red”,
yaxt=”n”, lty=3, ann=FALSE)
lines(x, z, type=”b”, pch=22, col=”blue”, lty=2)
axis(2, at=x, labels=x, col.axis=”red”, las=2)
axis(4, at=z, labels=round(z, digits=2),
col.axis=”blue”, las=2, cex.axis=0.7, tck=-.01)
mtext(“y=1/x”, side=4, line=3, cex.lab=1, las=2, col=”blue”)
title(“Пример осей” ,
xlab=”значение переменной X”,
ylab=”Y=X”)
par(opar)
```

18. Изучить самостоятельно добавление Легенды к диаграмме
19. По экспериментальным данным провести построение 3 - 4 различных диаграмм
20. Используя изученные функции и команды провести модификацию построенных графиков.

Содержание отчета

Отчет по выполняемой лабораторной работе выполняется каждым студентом индивидуально на листах формата А4 в рукописном или машинном варианте исполнения и должен содержать:

- название работы;
- цель и задачи исследований;
- набор эксперемнтальных данных;
- построенные диаграммы;
- программный код с комментариями;
- выводы по работе.

Контрольные вопросы

7. Принцип построения диаграмм в языке R.
8. Функции изменения графических параметров диаграмм
9. Команды Добавление текста,
10. Команды настройки параметров осей
11. Команды добавления условных обозначений.

Лабораторная работа №2.3
Корреляционный и регрессионный анализ данных. Исследование тесноты взаимосвязей данных в среде R

Цель:

– исследовать возможности языка R для определения тесноты взаимосвязей экспериментальных данных;

Время: 2 часа

Лабораторное оборудование: персональные компьютеры, выход в сеть Internet, RStudio.

Краткие теоретические сведения

Процедура поиска предполагаемой зависимости между различными числовыми совокупностями обычно включает следующие этапы:

− установление значимости связи между ними;

− возможность представления этой зависимости в форме математиче-ского выражения (уравнения регрессии).

Первый этап в указанном статистическом анализе касается выявления так называемой корреляции, или корреляционной зависимости. Корреляция рассматривается как признак, указывающий на взаимосвязь ряда числовых последовательностей. Иначе говоря, корреляция характеризует силу взаимосвязи в данных. Если это касается взаимосвязи двух числовых массивов x_i и y_i, то такую корреляцию называют парной.

При поиске корреляционной зависимости обычно выявляется вероятная связь одной измеренной величины x (для какого-то ограниченного диапазона ее изменения, например от x_1 до x_n) с другой измеренной величиной y (также изменяющейся в каком-то интервале $y_1 \ldots y_n$). В таком случае мы будем иметь дело с двумя числовыми последовательностями, между которыми и надлежит установить наличие статистической (корреляционной) связи. На этом этапе пока не ставится задача определить, является ли одна из этих случайных величин функцией, а другая – аргументом. Отыскание количественной зависимости между ними в форме конкретного аналитического выражения $y = f(x)$ − это задача уже другого анализа, регрессионного.

Таким образом, корреляционный анализ позволяет сделать вывод о силе взаимосвязи между парами данных x и y, а регрессионный анализ используется для прогнозирования одной переменной (y) на основании другой (x). Иными словами, в этом случае пытаются выявить причинно-следственную связь между анализируемыми совокупностями. Схематическое изображение изложенных соображений представлено на рис.1.

Рисунок.1. Схематическое пояснение сути корреляционного и регрессионного анализов

Принято различать два вида связи между числовыми совокупностями – это может быть **функциональная** зависимость или же **статистическая** (случайная). При наличии функциональной связи каждому значению воздействующего фактора (аргумента) соответствует строго определенная величина другого показателя (функции), т.е. изменение результативного признака всецело обусловлено действием факторного признака.

По своему характеру корреляционные связи – это соотносительные связи.

Такая зависимость графически изображается в виде экспериментальных точек, образующих поле рассеяния, или, как принято говорить, поле корреляции (рис.2).

Рисунок 2. Поле корреляции.

Следовательно, такие двумерные данные можно анализировать с использованием диаграммы рассеяния в координатах «x – y», которая дает визуальное представление о взаимосвязи исследуемых совокупностей. Для количественной оценки существования связи между изучаемыми совокупностями случайных величин используется специальный статистический показатель – **коэффициент корреляции** r.

Если предполагается, что эту связь можно описать линейным уравнением типа $y = a + bx$ (где a и b − константы), то принято говорить о существовании линейной корреляции. Коэффициент r − это безразмерная величина, она может меняться от 0 до ±1. Чем ближе значение коэффициента к единице (неважно, с каким знаком), тем с большей уверенностью можно утверждать, что между двумя рассматриваемыми совокупностями переменных существует линейная связь. Иными словами, значение какой-то одной из этих случайных величин (y) существенным образом зависит от того, какое значение принимает другая (x). Если окажется, что $r = 1$ (или −1), то имеет место классический случай чисто функциональной зависимости (т.е. реализуется идеальная взаимосвязь).

При анализе двумерной диаграммы рассеяния можно обнаружить различные взаимосвязи. Простейшим вариантом является линейная взаимосвязь, которая выражается в том, что точки размещаются случайным образом вдоль прямой линии. Диаграмма свидетельствует об отсутствии взаимосвязи, если точки расположены случайно, и при перемещении слева направо невозможно обнаружить какой-либо уклон (ни вверх, ни вниз). Если точки на ней группируются вдоль кривой линии, то диаграмма рассеяния характеризуется нелинейной взаимосвязью.

Методы определения корреляционной связи

Корреляцию и регрессию принято рассматривать как совокупный процесс статистического исследования, поэтому их использование в статистике часто именуют корреляционно-регрессионным анализом.

Чтобы выявить наличие качественной корреляционной связи между двумя исследуемыми числовыми наборами экспериментальных данных, существуют различные методы, которые принято называть элементарными. Ими могут быть приемы, основанные на следующих операциях:

− параллельном сопоставлении рядов;

− построении корреляционной и групповой таблиц;

− графическом изображении с помощью поля корреляции.

Другой метод, более сложный и статистически надежный, − это количественная оценка связи посредством расчета коэффициента корреляции и его статистической проверки.

Корреляция моментов Пирсона. Если форма распределения анализируемых признаков не очень сильно отличается от нормальной и отсутствуют выбросы, рассчитывают коэффициент **корреляции Пирсона** (часто называемый просто коэффициентом корреляции):

$$r_{xy} = \frac{n\sum xy - \sum x \sum y}{\sqrt{\left[n\sum x^2 - \left(\sum x\right)^2\right] \cdot \left[n\sum y^2 - \left(\sum y\right)^2\right]}}$$

где x - значение факторного признака;

y - значение результативного признака;

n - число пар данных.

Коэффициент корреляции величина относительная; он принимает значение от минус единицы до плюс единицы, т.е. $-1 < r < 1$.

При $r > 0$ связь оценивается, как прямая, при $r < 0$ – обратная.

При $r = 0$ – связь отсутствует, при $|r| = 1$ – связь функциональная

Сила связи оценивается:

при $|r| < 0{,}3$ – как слабая,

при $0{,}3 < [r| < 0{,}7$ – умеренная,

при $|r| > 0{,}7$ – сильная.

Следует помнить, что корреляция между величинами x и y не обя-зательно отражает причинно-следственные связи между ними.

Ранговые коэффициенты корреляции. Непараметрические, или ранговые коэффициенты корреляции Спирмена или Кендалла, рассчитывают в следующих случаях:

– форма распределения отличается от нормальной, например, скошена в ту или иную сторону;

– есть значительные выбросы (которые отражают не ошибки измерений или регистрации данных, а их реальные особенности);

– шкала измерений не количественная, а порядковая;

– небольшой размер выборки.

При вычислении коэффициента корреляции **Спирмена** (r_s) величины x сортируют по возрастанию и ранжируют. Равные величины получают средние значения из рангов, которые получили бы эти значения без ограничения. Аналогично присваивают ранги величинам y. Находят разности рангов x_i и y_i, обозначаемые d_i.

$$r_s = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}.$$

Коэффициент корреляции **Кендалла** (τ) рассчитывают по следующему алгоритму:

1. Значения x ранжируют по возрастанию.
2. Значения y располагают соответственно x и ранжируют.
3. Для каждого ранга y_i определяют число следующих за ним значений рангов, больших y_i. Суммируют и получают n_c – число согласованных пар (от англ. *concordant*), или последовательностей.

4. Для каждого значения y_i определяют число следующих за ним рангов, меньших y_i. Суммируют и получают n_d –число рассогласованных пар (от англ. *discordant*), или инверсий.

5. Рассчитывают τ по формуле:

$$\tau = \frac{2(n_c - n_d)}{n(n-1)}$$

Статистическая значимость коэффициента корреляции. Уровень статистической значимости коэффициента корреляции (Пирсона, Спирмена или Кендалла) зависит от числа наблюдений и может быть оценен с помощью следующей статистики:

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}}$$

где r – коэффициент корреляции, n – число наблюдений.

При этом проверяются следующие статистические гипотезы:

H_0: Корреляция между переменными не отличается от нуля.

H_1: Корреляция между переменными достоверно отличается от нуля.

При уровне статистической значимости $p<0,05$ отвергается нулевая гипотеза и считается, что связь между изучаемыми переменными действительно существует.

Функции R для проведения корреляционного анализа

Функция `cor` ()

Для расчета коэффициентов корреляции применяют функцию:

`cor (x, use=, method= ).`

Функция возвращает матрицу корреляций между указанными переменными.

Аргументы

`x` – набор данных, между которыми вычисляются корреляции (можно указать имена переменных рабочего файла в кавычках);

`use` – обработка пропусков.

Возможные значения:

`all.obs` – без пропусков (пропущенные данные вызовут ошибку);

`complete.obs` –построчная обработка пропусков;

`pairwise.complete.obs` – попарная обработка пропусков;

`method` – вид корреляции. Возможные значения:

`pearson` – корреляция Пирсона;

`spearman` – корреляция Спирмена;

`kendall` – корреляция Кендалла.

Функция `cor.test` ()

Для оценки уровня значимости коэффициентов корреляции применется функция:

`cor.test (x, y, alternative=, method= ).`

Аргументы

`x, y` – набор данных, должны быть одинаковой длины;

`alternative` –альтернативная гипотеза (двусторонний или односторонний тест). Возможные значения:

`two.sided`;

`greater`;

`less`;

`method` – вид тестируемой корреляции. Возможные значения:

`pearson` – корреляция Пирсона;
`spearman` – корреляция Спирмена;
`kendall` – корреляция Кендалла.

Функции R для графической интерпретации корреляционного анализа

Для визуального исследования зависимости между двумя переменными используют двумерные диаграммы рассеяния, или графики разброса.

Функция `scatterplot` ()

Для создания графика разброса между двумя переменными применяют функцию:

`scatterplot(formula, data, xlab, ylab, legend.title, ellipse, reg.line, smooth).`

Аргументы

`formula` – «формула» для построения графика, применяют в форме y~x или y~x | z, где z – фактор, подразделяющий выборку на подгруппы;

`data` – массив данных, по которому строится график разброса;

`xlab` – название горизонтальной оси;

`ylab` – название вертикальной оси;

`legend.title` – заголовок легенды;

`ellipse` – при значении TRUE вместо точек на графике отображаются корреляционные эллипсы;

reg.line – отображает линию линейной регрессии при значении TRUE и не отображает её при значении FALSE;

smooth – отображает кривую нелинейной регрессии при значении TRUE и не отображает её при значении FALSE.

При необходимости построить матрицу парных графиков по нескольким переменным можно воспользоваться функцией: `scatterplot.matrix ()`.

Аргументы данной функции во многом аналогичны таковым функции `scatterplot`(), иначе пишется «формула» графика: ~x1+x2+x3… Помимо указанных функций, код которых генерирует оболочка R commander, существует функция `pairs()`, которая также создаёт графики разброса.

Пример матрицы парных графиков гематологических показателей приведён на рис. 1. По диагонали представлены имена анализируемых переменных. Каждая точка отражает одно наблюдение, её координаты определяются значениями двух переменных. Выше и ниже диагонали с именами переменных расположены одни и те же пары переменных, но по разным осям.

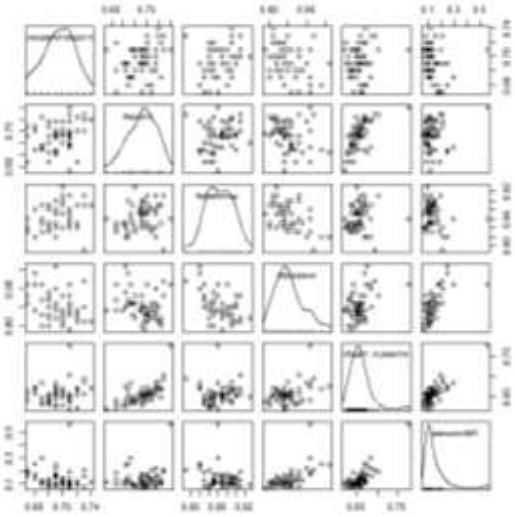


Рисунок 1. Диаграммы рассеяния

Если переменные тесно и линейно связаны, то множество точек данных принимает форму узкого эллипса или почти прямой.

Диаграммы рассеяния предоставляют исследователю больше информации, чем простое значение коэффициента корреляции. Они позволяют:

– выявить отсутствие однородности в выборке (например, наличие подгрупп с разным характером взаимосвязи);

– найти выбросы, или нетипичные данные, которые искусственным образом могут значительно увеличить или уменьшить коэффициент корреляции Пирсона;

– обнаружить нелинейный характер взаимосвязи.

Таким образом, перед проведением корреляционного анализа желательно анализировать графики разброса, с помощью которых можно подобрать оптимальный срез данных для исследования (т.е. выделить определённые подгруппы или, наоборот, объединить разные подгруппы в одну, исключить выскакивающие наблюдения) и применить подходящий вид корреляции (Пирсона или его непараметрических аналогов – Спирмена или Кендалла).

Парная линейная регрессия

Регрессионный анализ решает следующие задачи:

– восстановление зависимости между исследуемыми переменными;

– прогноз зависимой переменной (переменной отклика) по известным независимым переменным (предикторам).

Регрессия бывает:

– по числу предикторов – парная и множественная;

– по форме зависимости – линейная и криволинейная.

Исходные данные для регрессионного анализа представляют собой таблицу (матрицу), в которой строки соответствуют объектам (испытуемым), а столбцы – переменным. Все переменные при этом должны быть измерены в количественной шкале. Одна из переменных определяется исследователем как зависимая, а остальные как независимые переменные.

Парная линейная регрессия в общем случае имеет вид: $y=b_0+b_1x$ Нахождение коэффициентов регрессии основано на методе наименьших квадратов (минимизация суммы квадратов отклонений эмпирических значений признака от теоретических, полученных по уравнению регрессии)

Функции R для построения линейной регрессии

Для построения линейной регрессии в пакете R можно воспользоваться функцией `lm()`:

`lm(formula, data, subset, weights, na.action).`

Аргументы

`formula` – символическое описание восстанавливаемой модели. Для парной линейной регрессии имеет вид $y{\sim}x$, для множественной – $y{\sim}x_1+x_2+x_3\ldots$;

`data` – источник данных;

`subset` – подмножество данных, участвующих в построении модели, необязательный параметр;

`weights` – вектор весов, может быть или NULL, или числовым;

`na.action` – обработка пропущенных данных (NA).

Трактовка результатов

Объект, возвращаемый функцией `lm`, имеет различные поля:

`Residuals` – остатки ($y_i–bx_i$), распределение по квартилям.

`Coefficients` – коэффициенты регрессии и их статистическая значимость:

`Estimate` – коэффициент регрессии b;

`Std. Error` – ошибка коэффициента регрессии b;

`t value` – статистика t для оценки уровня значимости коэффициента регрессии;

`Pr(>|t|)` – достигнутый уровень значимости коэффициента регрессии.

`Multiple R-squared` – коэффициент детерминации модели.
`Adjusted R-squared` – скорректированный коэффициент детерминации модели.
`F-statistic` – F-статистика для модели в целом.

Задание и порядок выполнения лабораторной работы №2.3

1. Запустить пакет R commander (Rcmdr). Дальнейшая работа будет проходит в данной графической оболочке, которая генерирует необходимый код через кнопочный интерфейс.

2. Загрузить данные для анализа из файла Данные.xlsx

Для загрузки данных в графической оболочке R commander выбрать следующие пункты меню:

Данные
Импорт данных
Из файла Excel.

3. Исследуйте взаимосвязь заданных индексов. Для этого воспользуйтесь корреляционным анализом:

Статистики
Итоги
Корреляционная матрица
В появившемся диалоговом окне укажите исследуемые переменные.

Выберите вначале корреляцию Пирсона, потом Спирмена. Проанализируйте полученную матрицу корреляций. Какие связи сильные, какие слабые, а какие умеренные? Для нескольких пар показателей найдите уровень значимости коэффициента корреляции.

Статистики
Итоги
Корреляционный тест

4. Проиллюстрируйте полученные результаты (для этих же переменных) на графиках разброса:

Графики
Матрица точечных графиков
Выбрать необходимые переменные.
Убрать галочки с «линии наименьших квадратов» и «сгладить линии».

5. Постройте уравнение зависимости индекса реального ВВП от индекса общественного развития. Для этого выберете следующие пункты меню в оболочке R commander:

Статистики
Подгонка моделей
Линейная регрессия

В появившемся диалоговом окне укажите: зависимая переменная – индекс реального ВВП, независимая – индекс общественного развития. Проанализируйте график остатков (наблюдаемое минус предсказанное регрессионной моделью значение). Одинаковы ли они на всём диапазоне значений предсказывающей переменной.

Модели
Графики
График Компонента + остаток

6. По таблице коэффициентов запишите полученное уравнение регрессии

7. Загрузите свои экспериментальные данные. Проведите корреляционный анализ всех данных. Проанализируйте полученную матрицу корреляций. Проиллюстрируйте полученные результаты (для этих же переменных) на графиках разброса.

8. Выберите 3 графика разброса. Укажите численное значение коэффициентов корреляции и их уровней значимости. Дайте содержательную оценку взаимосвязей: прямая или

обратная, слабая или сильная. Какие значения коэффициентов корреляции больше по абсолютному значению: Пирсона или Спирмена?

9. Постройте уравнение зависимости двух переменных. По таблице коэффициентов запишите полученное уравнение регрессии. Проанализируйте график остатков

Содержание отчета

Отчет по выполняемой лабораторной работе выполняется каждым студентом индивидуально на листах формата А4 в рукописном или машинном варианте исполнения и должен содержать:

- название работы;
- цель и задачи исследований;
- набор эксперемнтальных данных;
- выводы по п.п.7-9;
- программный код с комментариями;
- выводы по работе.

Контрольные вопросы

1. Корреляционный анализ.
2. Регрессионный анализ.
3. Методы определения корреляционной связи.
4. Коэффициенты корреляции
5. Функции R для проведения корреляционного анализа
6. Функции R для графической интерпретации корреляционного анализа.
7. Функции R для построения линейной регрессии

Лабораторная работа №2.4
Корреляционный и регрессионный анализ данных. Множественная линейная регрессия.

Цель:

– исследовать возможности языка R для построения множественной линейной регрессий.

Время: 2 часа

Лабораторное оборудование: персональные компьютеры, выход в сеть Internet, RStudio.

Краткие теоретические сведения

Когда в регрессионной модели есть одна зависимая и одна независимая переменная, такой подход называется простой линейной регрессией. Когда есть одна зависимая переменная, но в модель входят ее степени (например, X, X^2, X^3), это называется полиномиальной регрессией. Если есть больше одной независимой переменной, это называется множественной регрессией.

Множественная линейная регрессия

Если существует больше одной независимой переменной, простая линейная регрессия превращается во множественную линейную регрессию, а ход вычислений становится более сложным.

Множественная линейная регрессия позволяет изучить совместное воздействие нескольких независимых переменных на переменную отклика. Практическое применение двоякое: для предсказания переменной отклика и для определения интенсивности, с которой каждая независимая переменная линейно связана с зависимой. В общем случае уравнение множественной линейной регрессии имеет вид:

$$y = b_0 + b_1x_1 + b_2x_2... + b_kx_k .$$

Если существует больше одной независимой переменной, то регрессионные коэффициенты показывают, на сколько увеличится значение зависимой переменной при изменении данной независимой переменной на единицу при условии, что все остальные независимые переменные останутся неизменными.

Важным дополнительным условием является некоррелированность объясняющих переменных между собой (отсутствие мультиколлинеарности). Считается, что явление мультиколлинеарности наблюдается тогда, когда коэффициент корреляции между объясняющими переменными превышает по модулю 0,7.

Пошаговая множественная линейная регрессия

Существуют две методики построения множественной регрессии – пошаговая вперед и пошаговая назад.

Пошаговая вперед заключается в то, что первоначально строится модель с одной экзогенной переменной. Затем добавляется следующая и строится новая модель. Модели сравниваются и, в зависимости от того ухудшилась или улучшилась модель, введенная переменная либо остается в модели, либо заменяется на другую. Таким образом, перебираются различные комбинации экзогенных переменных, в результате получается наилучшая модель.

Пошаговая назад начинается с того, что рассчитывается множественная регрессия на всем множестве факторов. Затем построенная модель исследуется с точки зрения статистической

значимости модели в целом, статистической значимости коэффициентов регрессии, оценивается коэффициент детерминации. Затем из модели удаляется один из влияющих факторов.

Его выбор можно осуществить следующим образом:

1. Строится матрица парных коэффициентов корреляции между переменными.

2. Выбираются две экзогенные переменные, между которыми наибольший коэффициент парной корреляции.

3. Из этих двух переменных выбирается та, которая оказывает меньшее влияние на эндогенную переменную, и исключается из модели.

Затем строится новая модель, исследуется ее качество. Также проводится тест на лучшую из двух моделей: с меньшим или большим числом переменных. В конце получается наилучшая модель.

Результат применения метода пошаговой регрессии зависит от критериев включения или удаления переменных. При помощи функции `stepAIC()` из пакета MASS можно провести построение пошаговой регрессии с использованием точного критерия AIC (Akaike Information Criterion – информационный критерий Акаике).

При расчете этого критерия учитывается статистическое соответствие модели данным и число необходимых для достижения этого соответствия параметров. Предпочтение нужно отдавать моделям с **меньшими** значениями AIC, указывающими на хорошее соответствие данным при использовании меньшего числа параметров**.**

Мультиколлинеарность

Наибольшие затруднения в использовании аппарата множественной регрессии возникают при наличии мультиколлинеарности факторных переменных, когда более чем два фактора связаны между собой линейной зависимостью.

Мультиколлинеарностью для линейной множественной регрессии называется наличие линейной зависимости между факторными переменными, включёнными в модель.

Мультиколлинеарность – нарушение одного из основных условий, лежащих в основе построения линейной модели множественной регрессии.

Мультиколлинеарность можно выявить на начальном этапе моделирования (до построения регрессии). О ней могут свидетельствовать:

1. Большие (по абсолютной величине) парные коэффициенты корреляции между независимыми переменными.

2. Высокие (>10) значения коэффициента *VIF*.

Коэффициент *VIF* (variance inflation factor) характеризует силу мультиколлинеарности. Вычисление коэффициента выполняется с помощью функции `vif()`

Симптомами присутствия мультиколлинеарности в уже построенной модели являются:

1. Небольшое изменении исходных данных, приводит к существенному изменению оценок коэффициентов.

2. Каждая переменная в отдельности является незначимой, а уравнение в целом имеет высокий R^2 (коэффициент детерминации) и является значимым.

3. Оценки коэффициентов имеют неправильные с точки зрения экономической теории знаки или неоправданно большие значения

Задание и порядок выполнения лабораторной работы №2.4

Построение множественной линейной регрессии

10. Запустить пакет R commander (Rcmdr).
11. Загрузить данные для анализа из файла Данные.xlsx
12. Исследуйте парные взаимосвязи между всеми переменными.
13. Постройте диаграммы рассеяния

14. Проведите подгонку множественной регрессионной модель при помощи функции lm(). Для этого выберете следующие пункты меню в оболочке R commander:

Статистики

Подгонка моделей

Линейная регрессия

В появившемся диалоговом окне укажите: зависимая переменная – индекс реального ВВП, независимые – остальные индексы. По таблице коэффициентов запишите полученное уравнение регрессии, проанализируйте полученные коэффициенты регрессии.

Обозначения

• Первый столбец (без заголовка) – свободный член (`intercept`) и предикторы в модели;

• `Estimate` – коэффициент регрессии;

• `Std. Error` – стандартная ошибка коэффициента регрессии;

• `t value` – t-статистика, с помощью которой проверяют статистическую гипотезу отличия коэффициента регрессии от 0;

• `Pr(>|t|)` – уровень значимости предиктора, статистически значимые предикторы помечены звёздочками;

• `Residual standard error` – остаточная стандартная ошибка – показатель разброса возможных значений случайной ошибки;

• `Multiple R-squared` – коэффициент детерминации, или квадрат коэффициента множественной корреляции модели;

• `Adjusted R-squared` –скорректированный коэффициент детерминации;

• `F-statistic` – F-статистика, оценивающая значимость полученной модели множественной линейной регрессии в целом, и её уровень значимости.

15. Проанализируйте графики остатков (наблюдаемое минус предсказанное регрессионной моделью значение).

Модели

Графики

График Компонента + остаток

16. Загрузите свои экспериментальные данные.

17. Постройте множественную линейную регрессию. По таблице коэффициентов запишите полученное уравнение регрессии. Проанализируйте график остатков.

Построение пошаговой множественной регрессии

1. Запустить пакет R commander (Rcmdr).
2. Загрузить данные для анализа из файла Данные.xlsx
3. Выполнить пошаговое построение регрессии :

Модели

Ступенчаты выбор модели

В открывшемся окне выбрать коэффициент AIC, направление вперед

4. Сделать выводы по полученным результатам
5. Аналогично сделать пошаговое построение по направлению назад, сравнить с предыдущим построением, сделать выводы.
6. Проверить коэффициент VIF и сделать выводы

Модели

Числовая диагностика

Факторы влияющие на дисперсию

7. Загрузить свои экспериментальные данные
8. Выполнить пошаговое построение множественной регрессии по направлению назад (вперед) сравнить полученные результаты, провести проверку коэффициент VIF, сделать выводы.

Содержание отчета

Отчет по выполняемой лабораторной работе выполняется каждым студентом индивидуально на листах формата А4 в рукописном или машинном варианте исполнения и должен содержать:

- название работы;
- цель и задачи исследований;
- набор экспериментальных данных;
- выводы по построенной множественной регрессии;
- программный код с комментариями;
- выводы по работе.

Контрольные вопросы

1. Множественная регрессия
2. Пошаговая множественная регрессия
3. Мультиколлинеарность.

Лабораторная работа №3
Задача дисперсионного анализа. Методы дисперсионного анализа. Однофакторный дисперсионный анализ.

Цель:

- приобрести практические навыки в проведении дисперсионного анализа по экспериментальным данным
- исследовать возможности языка R для проведения дисперсионного анализа.

Время: 4 часа

Лабораторное оборудование: персональные компьютеры, выход в сеть Internet, RStudio.

Краткие теоретические сведения

Дисперсионный анализ разработан в 20-х годах XX века английским математиком и генетиком Рональдом Фишером для обработки результатов агрономических опытов по выявлению условий получения максимального урожая различных сортов сельскохозяйственных культур. Сам термин «дисперсионный анализ» Фишер употребил позднее.

В настоящее время дисперсионный анализ определяется как статистический метод, предназначенный для оценки влияния различных факторов на результат эксперимента, а также для последующего планирования аналогичных экспериментов.

Дисперсионный анализ (от латинского Dispersio – рассеивание / на английском Analysis Of Variance – ANOVA) применяется для исследования влияния одной или нескольких качественных переменных (факторов) на одну зависимую количественную переменную (отклик).

В основе дисперсионного анализа лежит предположение о том, что одни переменные могут рассматриваться как причины (факторы, независимые переменные): $f_1, \dots f_k$, а другие как следствия (зависимые переменные). Независимые переменные называют иногда регулируемыми факторами именно потому, что в эксперименте исследователь имеет возможность варьировать ими и анализировать получающийся результат.

Дисперсионный анализ особенно эффективен при изучении нескольких факторов. При классическом методе исследования варьируют только один фактор, а остальные оставляют постоянными. при этом для каждого фактора проводится серия наблюдений, не используемая при изучении других факторов. Кроме того, при таком методе исследования не удается определить взаимодействие факторов при одновременном их изменении. При дисперсионном анализе каждое наблюдение служит для одновременной оценки всех факторов и их взаимодействий.

Дисперсионный анализ состоит в выделении и оценке отдельных факторов, вызывающих изменчивость изучаемой случайной величины. Для этого производится разложение суммарной выборочной дисперсии на составляющие, обусловленные независимыми факторами. Каждая из этих составляющих представляет собой оценку дисперсии генеральной совокупности. Чтобы решить, значимо ли влияние данного фактора, необходимо оценить значимость соответствующей выборочной дисперсии в сравнении с дисперсией воспроизводимости, обусловленной случайными факторами. Проверка значимости оценок дисперсий проводится по критерию Фишера. Если рассчитанное значение критерия Фишера окажется меньше табличного, то влияние рассматриваемого фактора нет оснований считать значимым. Если же рассчитанное значение критерия Фишера окажется больше табличного, то рассматриваемый фактор влияет на изменчивость средних.

В дальнейшем будем полагать, что выполняются следующие допущения:

- случайные ошибки наблюдений имеют нормальное распределение;

- факторы влияют только на изменение средних значений, а дисперсия наблюдений остается постоянной;

- эксперименты равноточны.

Требование нормального распределения определяет выбор основных факторов при исследовании процесса методом дисперсионного анализа. Если нужно получить нормальное распределение выходной величины, к случайным желательно относится только те факторы, влияние которых на выходную величину очень мало. Исключение можно делать лишь для тех факторов, которые сами по себе (из каких-либо других соображений) дают нормальное распределение результатов.

Факторы рассматриваемые в дисперсионном анализе, бывают двух родов:

- со случайными уровнями;
- с фиксированными.

В первом случае предполагается, что выбор уровней производится из бесконечной совокупности возможных уровней и сопровождаются рандомизацией. При этом результаты эксперимента имеют большее значение, поскольку выводы по эксперименту можно распространить на всю генеральную совокупность. Если все уровни выбираются случайным образом, математическая модель эксперимента называется моделью со случайными уровнями факторов (случайная модель). Когда все уровни фиксированы, модель называется моделью с фиксированными уровнями. Когда часть факторов рассматривается на фиксированных уровнях, а уровни остальных выбираются случайным образом, модель называется моделью смешанного типа. Иногда отсутствие различие в критериях, применяемых для разных моделей, и единственное различие состоит в общности выводов, в других случаях существует различие в критериях.

Дисперсионный анализ может применяться в различных формах в зависимости от структуры исследуемого процесса; выбор соответствующей формы является обычно одной из главных трудностей в практическом применении анализа.

По числу факторов различают однофакторный и многофакторный дисперсионный анализ

Однофакторный дисперсионный анализ

Задачей дисперсионного анализа является изучение влияния одного или нескольких факторов на рассматриваемый признак.

Однофакторный дисперсионный анализ используется в тех случаях, когда есть в распоряжении три или более независимые выборки, полученные из одной генеральной совокупности путем изменения какого-либо независимого фактора, для которого по каким-либо причинам нет количественных измерений.

Для этих выборок предполагают, что они имеют разные выборочные средние и одинаковые выборочные дисперсии. Поэтому необходимо ответить на вопрос, оказал ли этот фактор существенное влияние на разброс выборочных средних или разброс является следствием случайностей, вызванных небольшими объемами выборок. Другими словами, если выборки принадлежат одной и той же генеральной совокупности, то разброс данных между выборками (между группами) должен быть не больше, чем разброс данных внутри этих выборок (внутри групп).

Пусть x_{ik} – i – элемент ($i=\overline{1,n_k}$) k-выборки ($k=\overline{1,m}$), где m – число выборок, nk – число данных в k-выборке. Тогда $\overline{x_{ik}}$ – выборочное среднее k-выборки определяется по формуле

$$\overline{x}_k = \frac{1}{n_k}\sum_{i=1}^{nk} x_{ik}.$$

Общее среднее вычисляется по формуле

$$\bar{x} = \frac{1}{n}\sum_{k=1}^{m}\sum_{i=1}^{nk} x_{ik},$$

где $n = \sum_{k=1}^{m} n_k$.

Основное тождество дисперсионного анализа имеет следующий вид:

$$Q = Q_1 + Q_2,$$

где Q_1 – сумма квадратов отклонений выборочных средних $\overline{x_{ik}}$ от общего среднего $\overline{x}$ (сумма квадратов отклонений между группами);

Q2 – сумма квадратов отклонений наблюдаемых значений x_{ik} от выборочной средней $\overline{x_k}$ (сумма квадратов отклонений внутри групп);

Q – общая сумма квадратов отклонений наблюдаемых значений x_{ik} от общего среднего $\overline{x}$.

Расчет этих сумм квадратов отклонений осуществляется по следующим формулам:

$$Q_1 = \sum_{k=1}^{m} n_k (\bar{x}_k - \bar{x})^2 = \sum_{k=1}^{m} n_k \bar{x}_k^2 - n\bar{x}^2,$$

$$Q_2 = \sum_{k=1}^{m}\sum_{i=1}^{nk} (x_{ik} - \bar{x}_k)^2 = \sum_{k=1}^{m}\sum_{i=1}^{nk} x^2{}_{ik} - \sum_{k=1}^{m} n_k \bar{x}_k^2.$$

В качестве критерия необходимо воспользоваться критерием Фишера:

$$F = \frac{Q_1/(m-1)}{Q_2/(n-m)}.$$

Если расчетное значение критерия Фишера будет меньше, чем табличное значение нет оснований считать, что независимый фактор оказывает влияние на разброс средних значений, в противном случае, независимый фактор оказывает существенное влияние на разброс средних значений (α– уровень значимости, уровень риска, обычно для экономических задач α=0,05).

Недостаток однофакторного анализа: невозможно выделить те выборки, которые отличаются от других. Для этой цели необходимо использовать метод Шеффе или проводить парные сравнения выборок.

Задание и порядок выполнения лабораторной работы №3

Дисперсионный анализ в MS EXCEL

1. Создать файл с исходными данными (варианты заданий представлены в Приложении А).
2. Запустить “Пакет анализа”.

В системе электронных таблиц Microsoft Excel имеется набор инструментов для анализа данных, называемый «Пакет анализа», который может быть использован для решения сложных статистических задач. Для использования одного из этих инструментов указать входные данные

и выбрать параметры; анализ будет проведен с помощью подходящей статистической макрофункции, и результаты будут представлены в выходном диапазоне.

В меню Сервис (Данные) выберите команду Анализ данных. Если такая команда отсутствует в меню Сервис (Данные), то необходимо установить в Microsoft Excel пакет анализа данных.

Установка производится следующим образом. В меню Сервис (Файл → Параметры) выберите команду Надстройки. Если в списке надстроек нет пакета анализа данных, то нажмите кнопку “Обзор” и задайте диск, каталог и имя файла для надстройки “Пакет анализа”, или запустите программу установки Microsoft Excel. Установите флажок “Пакет анализа” (надстройки, установленные в Microsoft Excel, остаются доступными, пока не будут удалены).

Выберите необходимую строку в списке “Инструменты анализа”.

Введите входной и выходной диапазоны, затем выберите необходимые параметры. Для использования инструментов анализа исследуемые данные следует представить в виде строк или столбцов на листе. Совокупность ячеек, содержащих анализируемые данные, называется входным диапазоном.

3. Провести однофакторный дисперсионный анализ.

В меню Сервис выбираем команду Анализ данных.

В списке инструментов статистического анализа выбираем Однофакторный дисперсионный анализ (Рисунок 1).

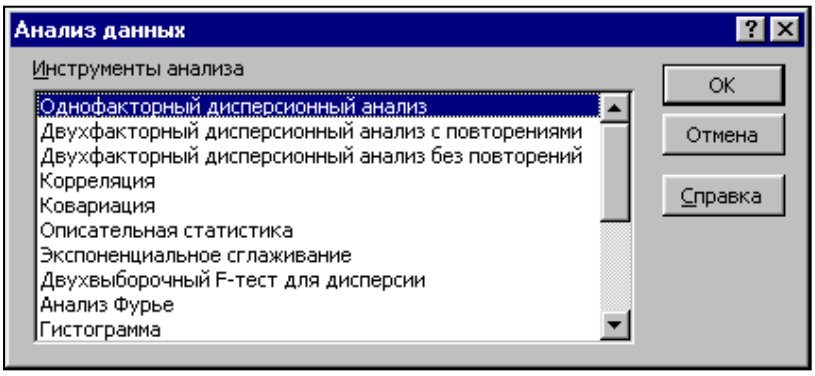


Рисунок 1 – Выбор инструмента анализа

В диалоговом окне режима (Рисунок 2) указываем входной интервал, способ группирования, выходной интервал, метки в первой строке/ Метки в первом столбце, альфа (уровень значимости).

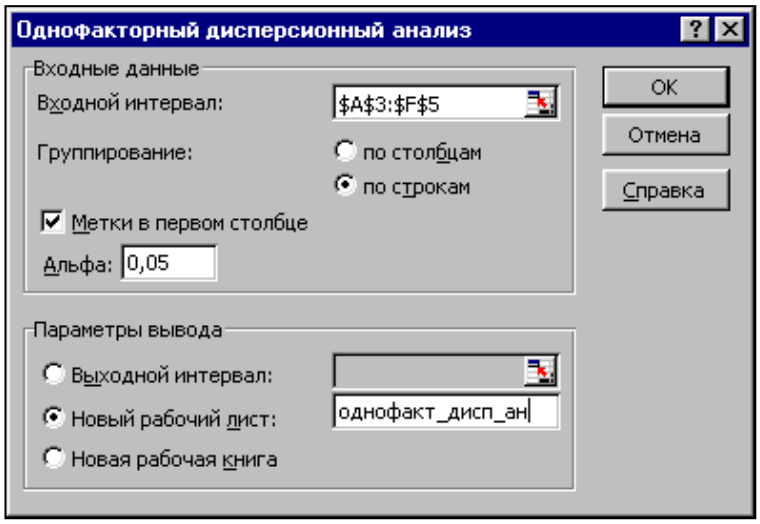


Рисунок 2 – Диалоговое окно однофакторного дисперсионного анализа

Входной диапазон – это ссылка на ячейки, содержащие анализируемые данные. Ссылка должна состоять как минимум из двух смежных диапазонов данных, организованных в виде столбцов или строк. Входной интервал можно задать при помощи мыши, или набрать на клавиатуре.

Группирование. Установите переключатель в положение “по столбцам” или “по строкам” в зависимости от расположения данных во входном диапазоне.

Метки в первой строке/ Метки в первом столбце. Установите переключатель в положение “Метки в первой строке”, если первая строка во входном диапазоне содержит названия столбцов. Установите переключатель в положение “Метки в первом столбце”, если названия строк находятся в первом столбце входного диапазона. Если входной диапазон не содержит меток, то необходимые заголовки в выходном диапазоне будут созданы автоматически.

Выходной диапазон. Введите ссылку на ячейку, расположенную в левом верхнем углу выходного диапазона. Размеры выходной области будут рассчитаны автоматически, и соответствующее сообщение появится на экране в том случае, если выходной диапазон занимает место существующих данных или его размеры превышают размеры листа.

Новый лист. Установите переключатель, чтобы открыть новый лист в книге и вставить результаты анализа, начиная с ячейки A1. Если в этом есть необходимость, введите имя нового листа в поле, расположенном напротив соответствующего положения переключателя.

Новая книга. Установите переключатель, чтобы открыть новую книгу и вставить результаты анализа в ячейку A1 на первом листе в этой книге.

В результате обработки данных получили следующее:

	A	B	C	D	E	F	G
1	Однофакторный дисперсионный анализ						
2							
3	ИТОГИ						
4	*Группы*	*Счет*	*Сумма*	*Среднее*	*Дисперсия*		
5	I группа (контр.)	5	1673	334,6	56,8		
6	II группа	5	1812	362,4	220,8		
7	III группа	5	1885	377	276,5		
8							
9	ANOVA						
10	*Источник вариации*	*SS*	*df*	*MS*	*F*	*P-Значение*	*F критическое*
11	Между группами	4640	2	2319,8	12,55983	0,0011415	3,885290312
12	Внутри групп	2216	12	184,7			
13							
14	Итого	6856	14				
15							
16							
17							

однофакт_дисп_ан / Лист1 / Лист2 / Лист3 / Лист4 / Лист5 /

Рисунок 3 – Результаты однофакторного дисперсионного анализа

Таблица ИТОГИ:

“Счет” – число повторностей. “Сумма” – сумма значений показателя по строкам. “Дисперсия” – частная дисперсия показателя.

Таблица ANOVA представляет результаты дисперсионного анализа однофакторного комплекса, в котором первая колонка “Источник вариации” содержит наименование дисперсий. Графа “*SS*” – это сумма квадратов отклонений, “*df*” – степень свободы, графа “*MS*” – средний квадрат, “*F*” – критерий фактического F – распределения. “*P – значение*” – вероятность того, что дисперсия, воспроизводимая уравнением, равна дисперсии остатков. Определяет вероятность того, что полученная количественная определенность взаимосвязи между факторами и результатом может считаться случайной. “*F - критическое*” – это значение F – теоретического, которое впоследствии сравнивается с F – фактическим.

4. Сформулировать выводы.

Пример:

Задание 0 варианта. Для проверки влияния методики обучения производственным навыкам на качество подготовки отбираются случайным образом из выпускников ПТУ четыре группы учеников, которые после окончания обучения показали следующие производственные результаты (Таблица 1).

Группа (методика)	Выработка, шт./день							Число учеников
1	60	80	75	80	85	70		6
2	75	66	85	80	70	80	90	7
3	60	80	65	60	86	75		6
4	95	85	100	80				4

Проверить существенность влияния методики обучения на производственные результаты учеников

Выполнение.

1. Создать файл с исходными данными (рис 4)

	Методика 1	Методика 2	Методика 3	Методика 4
1	60	75	60	95
2	80	66	80	85
3	75	85	65	100
4	80	80	60	80
5	85	70	86	
6	70	80	75	
7		90		

Рисунок 4 – Исходные данные.

2. Запустить “Пакет анализа” (рис 5)

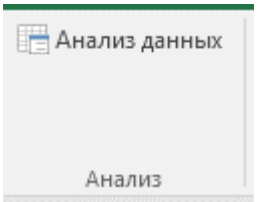


Рисунок 5 – Пакет анализа

3. Провести однофакторный дисперсионный анализ

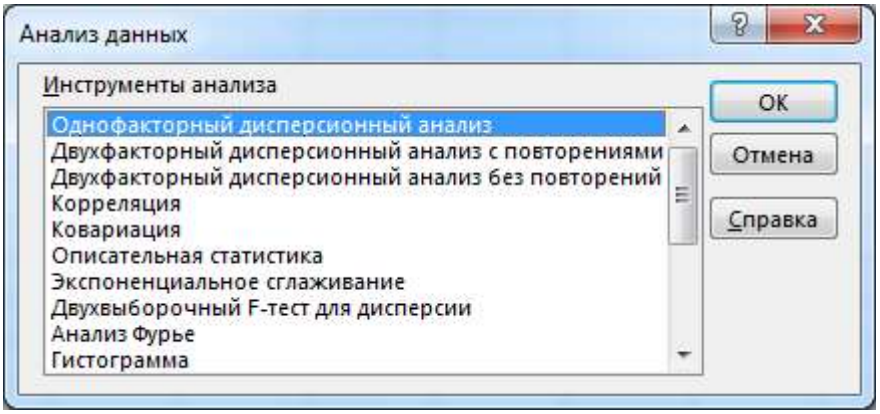


Рисунок 6 – Запуск процедуры «Однофакторный дисперсионный анализ)

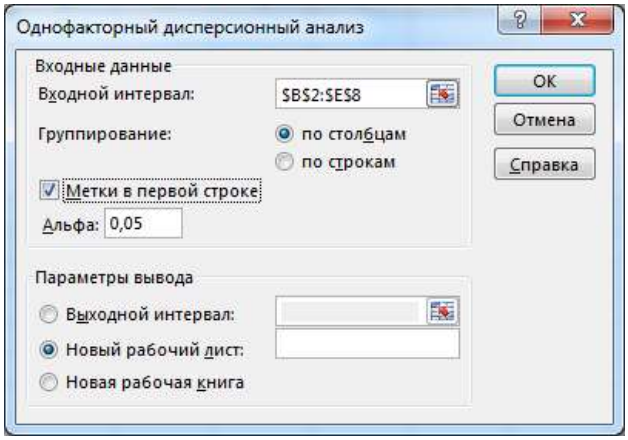


Рисунок 7 – Параметры однофакторного дисперсионного анализа

Однофакторный дисперсионный анализ						
ИТОГИ						
Группы	*Счет*	*Сумма*	*Среднее*	*Дисперсия*		
Столбец 1	6	450	75	80		
Столбец 2	7	546	78	69,66666667		
Столбец 3	6	426	71	120		
Столбец 4	4	360	90	83,33333333		
Дисперсионный анализ						
Источник вариации	*SS*	*df*	*MS*	*F*	*P-Значение*	*F критическое*
Между группами	917,7391	3	305,913	3,484620999	0,036142647	3,127350005
Внутри групп	1668	19	87,78947			
Итого	2585,739	22				

Рисунок 8 – Результирующая таблица выполненного дисперсионного анализа.

4. Сформулировать выводы

Таблица ИТОГИ:

Сравнение средних значений показывает, что Методика 4 (столбец 4) позволяет добиться лучшего результата, а Методика 3 наименее эффективна.

Назначение итоговой таблицы дисперсионного анализа проверить нулевую гипотезу H_0: об отсутствии значимого влияния уровней факторов на исследуемый отклик.

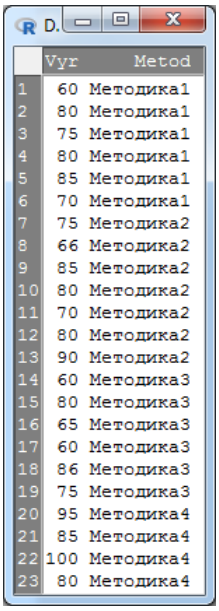
Сравнение F и Fкритическое показывает, что F > Fкритическое, следовательно отвергнута гипотеза H_0 и принята гипотеза H_1 и с вероятностью ошибки $\alpha = 0{,}05$ можно утверждать, что влияние рассматриваемого фактора на результативный признак существенно.

Дисперсионный анализ средствами языка R

1. Создать набор данных согласно варианту
   Набор данных создается или с использованием среды Rcmdr или импортируется.
2. Провести однофакторный дисперсионный анализ в среде Rcmdr
   Статистика
   Средние
   Одномерный дисперсионный анализ
3. По результатам дисперсионного анализа сформулировать выводы.
4. Построить диаграмму, отображающую средние значения и их доверительные интервалы для каждой группы
   График
   График средних

Пример: Задание с предыдущего примера.

1. Создать набор данных



	Vyr	Metod
1	60	Методика1
2	80	Методика1
3	75	Методика1
4	80	Методика1
5	85	Методика1
6	70	Методика1
7	75	Методика2
8	66	Методика2
9	85	Методика2
10	80	Методика2
11	70	Методика2
12	80	Методика2
13	90	Методика2
14	60	Методика3
15	80	Методика3
16	65	Методика3
17	60	Методика3
18	86	Методика3
19	75	Методика3
20	95	Методика4
21	85	Методика4
22	100	Методика4
23	80	Методика4

Рисунок 9 – Набор данных в среде Rcmdr

2. Провести однофакторный дисперсионный анализ в среде Rcmdr

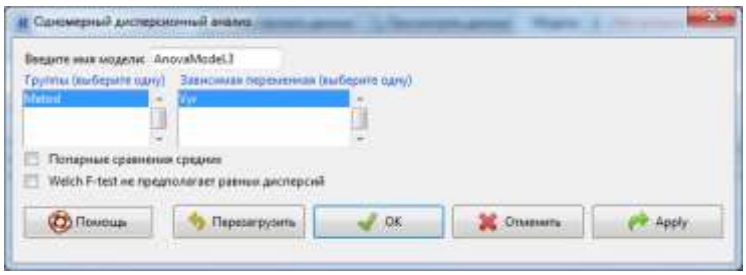


Рисунок 10 - Одномерный дисперсионный анализ в среде Rcmdr

```
> AnovaModel.3 <- aov(Vyr ~ Metod, data=Dataset)

> summary(AnovaModel.3)
            Df Sum Sq Mean Sq F value Pr(>F)
Metod        3  917.7  305.91   3.485 0.0361 *
Residuals   19 1668.0   87.79
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> with(Dataset, numSummary(Vyr, groups=Metod, statistics=c("mean", "sd")))
          mean        sd data:n
Методика1   75  8.944272      6
Методика2   78  8.346656      7
Методика3   71 10.954451      6
Методика4   90  9.128709      4
```

Рисунок 11 – Результирующая таблица

3. По результатам дисперсионного анализа сформулировать выводы

Строка, обозначенная как *Metod*, соответствует источнику дисперсии в данных, связанному с действием изучаемого экспериментального фактора/

Строка, обозначенная как *Residuals*, характеризует внутригрупповую дисперсию.

В столбце *F value* представлено рассчитанное по имеющимся данным значение F-критерия, он равен 3,485. В столбце *Pr(>F)* представлена вероятность получить F-значение, равное или превышающее то значение, которое в действительности рассчитали по имеющимся выборочным данным (при условии, что нулевая гипотеза верна). Как видно, эта вероятность не

высокая, равна 0,0361 (3,61%). Не превышает 5%-ный уровень значимости, в связи с чем мы заключаем, что нулевая гипотеза не верна. Таким образом, можно утверждать, что экспериментальные условия оказали существенное влияние на результативный признак.

Сравнение средних значений показывает, что Методика 4 позволяет добиться лучшего результата, а Методика 3 наименее эффективна.

4. Построить диаграмму, отображающую средние значения и их доверительные интервалы для каждой группы

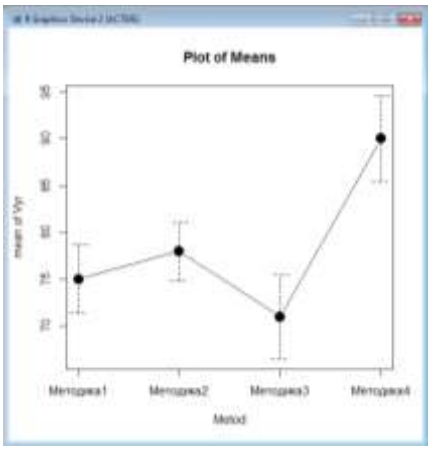


Рисунок 12 – Диаграмма, отображающая средние значения и их доверительные интервалы

Данная диаграмма визуально подтверждает, что Методика 4 позволяет добиться лучшего результата, а Методика 3 наименее эффективна

Контрольные вопросы

1. Факторы в дисперсионном анализе.
2. Цель применения дисперсионного анализа.
3. Различия между однофакторным и многофакторным дисперсионным анализом.
4. Три основные математические допущения дисперсионного анализа.
5. Внутригрупповая дисперсия в дисперсионном анализе.
6. Статистическая значимость F-отношения.
7. Функции для выполнения дисперсионного анализа на языке R.

Варианты заданий

Задача 1

При исследовании влияния стажа работы на производительность труда (количество деталей в день) в одном из цехов завода получен следующий однофакторный дисперсионный комплекс (таблица А1):

Таблица А.1.

Номер наблюдения	Стаж работы рабочих (лет)			
	До 5	5-10	10-15	15-20
Вариант 1				
1	155	154	153	164
2	153	158	162	162
3	149	157	164	163
4	150	161	163	
5			167	
Вариант 2				
1	147	149	155	160
2	149	150	149	163
3	153	152	156	166
4		148	161	
5			160	
Вариант 3				
1	155	159	158	168
2	158	161	166	167
3	156	164	163	164
4	154	158	165	166
5		163		
Вариант 4				
1	172	175	177	183
2	170	178	183	176
3	169	171	181	182
4		169	180	179
5		174		

Задача 2

В процессе исследования влияния цены за единицу продукции на объем продаж (шт.) в месяц были получены следующие результаты (таблица А2):

Таблица А2

Номер наблюдения	Цена за единицу продукции (руб.)			
	1000-1100	1100-1200	1200-1300	1300-1500
1	215	218	214	211
2	221	214	217	210
3	222	220	210	208
4	219	221		209
5		213		
Вариант 5				
1	267	266	262	264
2	270	271	265	265
3	275	272	267	260
4		265	268	259

5				261
Вариант 6				
1	310	311	308	299
2	314	309	307	287
3	311	305	300	301
4		307		300
Вариант 7				
1	56	55	49	44
2	58	52	51	43
3	55	53	45	39
4	59	48	41	
5		50	46	
Вариант 8				
1	97	85	89	79
2	93	88	83	81
3	96	90	85	80
4		94		82

Задача 3. Дана однофакторная таблица зависимости объема выручки (млн. руб.) от расходов на рекламу (тыс. руб.) (таблица А3):

Таблица А3

Номер исследования	Расходы на рекламу (тыс. руб.)			
	150-200	200-250	250-300	300-400
Вариант 9				
1	6,3	6,9	6,8	6,7
2	5,6	5,7	7,1	7,3
3	7,2	6,8	7,0	6,9
4	4,7		6,5	
Вариант 10				
1	7,1	7,4	7,5	7,7
2	7,3	7,9	7,7	7,9
3	7,7	8,4	7,2	7,8
4		7,6	7,8	8,3
5				8,0
Вариант 11				
1	6,6	6,0	8,4	8,7
2	5,9	6,8	7,5	7,8
3	6,4	7,4	6,9	7,1
4	7,1	8,1	7,3	7,6
5			7,7	
Вариант 12				
1	5,2	5,8	5,7	6,4
2	5,9	5,7	6,7	5,9
3	5,6	6,2	6,4	6,8
4		6,4	6,8	

Задача 4. Туристическими агентствами исследовалось влияние расстояния от пляжа (км) на наполняемость гостиниц (%). Были получены следующие данные (таблица А4):

Таблица А4

Номер исследования	Расстояние до пляжа (км)			
	До 1	1-2	2-4	4-6
Вариант 13				
1	99	98	96	89
2	98	97	94	90
3	100	99	95	93
4		97	94	92
5		96		91
Вариант 14				
1	98	97	98	90
2	97	96	95	87
3	99	94	94	94
4		99	96	88
5			97	
Вариант 15				
1	100	98	94	94
2	97	99	100	95
3	98	96	93	89
4		98	92	
5			90	
Вариант 16				
1	98	99	89	88
2	99	96	94	89
3	97	93	93	95
4		94	91	96
5			87	

Задача 5. Исследовалось влияние диеты на количество сброшенных килограммов за один месяц. Были получены следующие данные (таблица А5):

Таблица А5

Номер исследования	Номер диеты			
	1	2	3	4
Вариант 17				
1	3,2	4,5	3,3	4,1
2	1,6	3,4	5,4	1,7
3	2,3	1,8	2,9	3,9
4		1,9	3,0	
Вариант 18				
1	3,6	2,0	4,2	6,2
2	5,1	4,3	6,0	5,7
3	4,7	5,0	3,7	3,3
4	1,9	3,8	5,3	4,1
5			4,6	
Вариант 19				
1	2,2	4,2	5,5	5,1
2	4,1	4,3	4,7	4,3
3	2,6	5,6	3,9	6,0
4		3,9	5,0	4,2
5				3,8

Вариант 20				
1	5,2	6,2	4,8	7,1
2	4,6	6,6	4,9	7,6
3	5,1	5,1	5,7	4,5
4	5,5		4,3	5,8
5				5,9

Задача 6. На предприятии были проведены исследования влияния периода реализации продукции а объем выручки (млн. руб.) (таблица А6):

Таблица А6

Номер исследования	Период проведения исследования			
	I квартал	II квартал	III квартал	IV квартал
Вариант 21				
1	150	156	149	155
2	154	154	152	166
3	155	161	148	162
4		155		
Вариант 22				
1	168	166	154	164
2	167	159	159	163
3	159	163	151	157
4	156	158		160
5		167		
Вариант 23				
1	191	200	197	187
2	198	210	211	195
3	201	199	204	188
4		189	208	
5			194	
Вариант 24				
1	179	191	189	189
2	181	189	182	197
3	183	196	176	199
4	189	182		201
5		188		

Задача 7. Исследовалось влияние количества осадков за год на урожайность пшеницы (т/га). Получены следующие данные (таблица А7):

Таблица А7

Номер исследования	Количество осадков (мм)			
	250-260	260-270	270-280	280-290
Вариант 25				
1	33	29	34	34
2	31	30	37	36
3	32	28	33	38
4		33		
Вариант 26				
1	28	29	33	37
2	29	32	35	38
3	31	33	31	34
4		27	36	

5			32	
Вариант 27				
1	39	28	31	32
2	30	26	33	37
3	27	31	28	35
4		32	35	31
5			30	30
Вариант 28				
1	28	30	29	33
2	26	29	34	34
3	29	34	31	37
4			33	36
5			30	

Лабораторная работа №4
Кластерный анализ. Основные этапы и задачи кластерного анализа данных.

Цель:

- Закрепить теоретические знания и приобрести практические навыки в проведении кластерного анализа по экспериментальным данным
- исследовать возможности языка R для проведения кластерного анализа.

Время: 6 часов

Лабораторное оборудование: персональные компьютеры, выход в сеть Internet, RStudio.

Краткие теоретические сведения

Кластеризация предназначена для разбиения совокупности объектов на однородные группы (кластеры или классы). Если данные выборки представить как точки в признаковом пространстве, то задача кластеризации сводится к определению "сгущений точек".

Цель кластеризации - поиск существующих структур.

Кластеризация является описательной процедурой, она не делает никаких статистических выводов, но дает возможность провести разведочный анализ и изучить "структуру данных".

Само понятие "кластер" определено неоднозначно: в каждом исследовании свои "кластеры". Переводится понятие кластер (cluster) как "скопление", "гроздь".

Кластер можно охарактеризовать как группу объектов, имеющих общие свойства.

Характеристиками кластера можно назвать два признака:

- внутренняя однородность;
- внешняя изолированность.

Термин кластерный анализ, впервые введенный Трионом (Tryon) в 1939 году, включает в себя более 100 различных алгоритмов.

Кластерный анализ позволяет сокращать размерность данных, делать ее наглядной.

Задачи кластерного анализа можно объединить в следующие группы:

1. Разработка типологии или классификации.
2. Исследование полезных концептуальных схем группирования объектов.
3. Представление гипотез на основе исследования данных.
4. Проверка гипотез или исследований для определения, действительно ли типы (группы), выделенные тем или иным способом, присутствуют в имеющихся данных.

Критерием для определения схожести и различия кластеров является расстояние между точками на диаграмме рассеивания. Это сходство можно "измерить", оно равно расстоянию между точками на графике.

Методов вычисления расстояний существует очень много (не забывайте, что дело происходит в многомерном пространстве). Наиболее широко употребляемыми методами для непрерывных переменных являются: эвклидово расстояние (рис. 1) – Euclidian distances (1) и манхеттеновское расстояние или расстояние городских кварталов – City-block (Manhattan) (2).

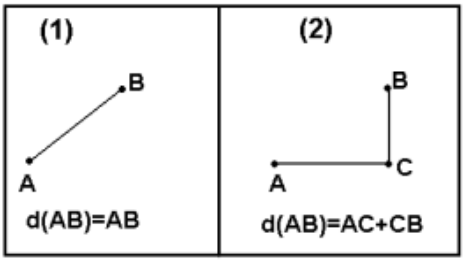


Рисунок – 1. Методов вычисления расстояний.

Наиболее распространенный способ - вычисление евклидова расстояния между двумя точками *i* и *j* на плоскости, когда известны их координаты *X* и *Y*:

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Методы кластерного анализа можно разделить на две группы:

– иерархические;
– неиерархические.

Суть иерархической кластеризации состоит в последовательном объединении меньших кластеров в большие или разделении больших кластеров на меньшие.

Иерархические агломеративные методы (Agglomerative Nesting, AGNES)

Эта группа методов характеризуется последовательным объединением исходных элементов и соответствующим уменьшением числа кластеров.

В начале работы алгоритма все объекты являются отдельными кластерами. На первом шаге наиболее похожие объекты объединяются в кластер. На последующих шагах объединение продолжается до тех пор, пока все объекты не будут составлять один кластер.

Иерархические дивизимные (делимые) методы (DIvisive ANAlysis, DIANA)

Эти методы являются логической противоположностью агломеративным методам. В начале работы алгоритма все объекты принадлежат одному кластеру, который на последующих шагах делится на меньшие кластеры, в результате образуется последовательность расщепляющих групп.

Принцип работы описанных выше групп методов в виде дендрограммы показан на рис. 2.

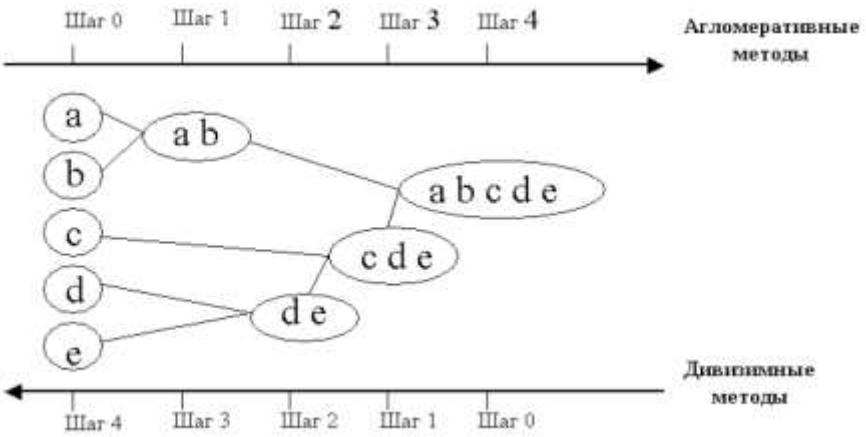


Рисунок – 2. Дендрограмма агломеративных и дивизимных методов.

Иерархические методы кластерного анализа используются при небольших объемах наборов данных.

Преимуществом иерархических методов кластеризации является их наглядность.

Иерархические алгоритмы связаны с построением дендрограмм (от греческого dendron - "дерево"), которые являются результатом иерархического кластерного анализа.

Дендрограмма описывает близость отдельных точек и кластеров друг к другу, представляет в графическом виде последовательность объединения (разделения) кластеров.

Дендрограмма (dendrogram) - древовидная диаграмма, содержащая *n* уровней, каждый из которых соответствует одному из шагов процесса последовательного укрупнения кластеров.

Дендрограмму также называют древовидной схемой, деревом объединения кластеров, деревом иерархической структуры.

Дендрограмма представляет собой вложенную группировку объектов, которая изменяется на различных уровнях иерархии.

Итеративные методы

При большом количестве наблюдений иерархические методы кластерного анализа не пригодны. В таких случаях используют неиерархические методы, основанные на разделении, которые представляют собой итеративные методы дробления исходной совокупности. В процессе деления новые кластеры формируются до тех пор, пока не будет выполнено правило остановки.

Такая неиерархическая кластеризация состоит в разделении набора данных на определенное количество отдельных кластеров. Существует два подхода. Первый заключается в определении границ кластеров как наиболее плотных участков в многомерном пространстве исходных данных, т.е. определение кластера там, где имеется большое "сгущение точек". Второй подход заключается в минимизации меры различия объектов.

Алгоритм k-средних (k-means)

Наиболее распространен среди неиерархических методов алгоритм k-средних, также называемый быстрым кластерным анализом.

В отличие от иерархических методов, которые не требуют предварительных предположений относительно числа кластеров, для возможности использования этого метода необходимо иметь гипотезу о наиболее вероятном количестве кластеров.

Алгоритм k-средних строит k кластеров, расположенных на возможно больших расстояниях друг от друга. Основной тип задач, которые решает алгоритм k-средних, - наличие предположений (гипотез) относительно числа кластеров, при этом они должны быть различны настолько, насколько это возможно. Выбор числа k может базироваться на результатах предшествующих исследований, теоретических соображениях или интуиции.

Общая идея алгоритма: заданное фиксированное число k кластеров наблюдения сопоставляются кластерам так, что средние в кластере (для всех переменных) максимально возможно отличаются друг от друга.

Описание алгоритма

1. Первоначальное распределение объектов по кластерам.

Выбирается число *k*, и на первом шаге эти точки считаются "центрами" кластеров.

Каждому кластеру соответствует один центр.

Выбор начальных центроидов может осуществляться следующим образом:

– выбор k-наблюдений для максимизации начального расстояния;
– случайный выбор k-наблюдений;
– выбор первых k-наблюдений.

В результате каждый объект назначен определенному кластеру.

2. Итеративный процесс.

Вычисляются центры кластеров, которыми затем и далее считаются покоординатные средние кластеров. Объекты опять перераспределяются.

Процесс вычисления центров и перераспределения объектов продолжается до тех пор, пока не выполнено одно из условий:

– кластерные центры стабилизировались, т.е. все наблюдения принадлежат кластеру, которому принадлежали до текущей итерации;
– число итераций равно максимальному числу итераций.

На рис. 3 приведен пример работы алгоритма k-средних для k, равного двум.

После получений 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 искажать 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.

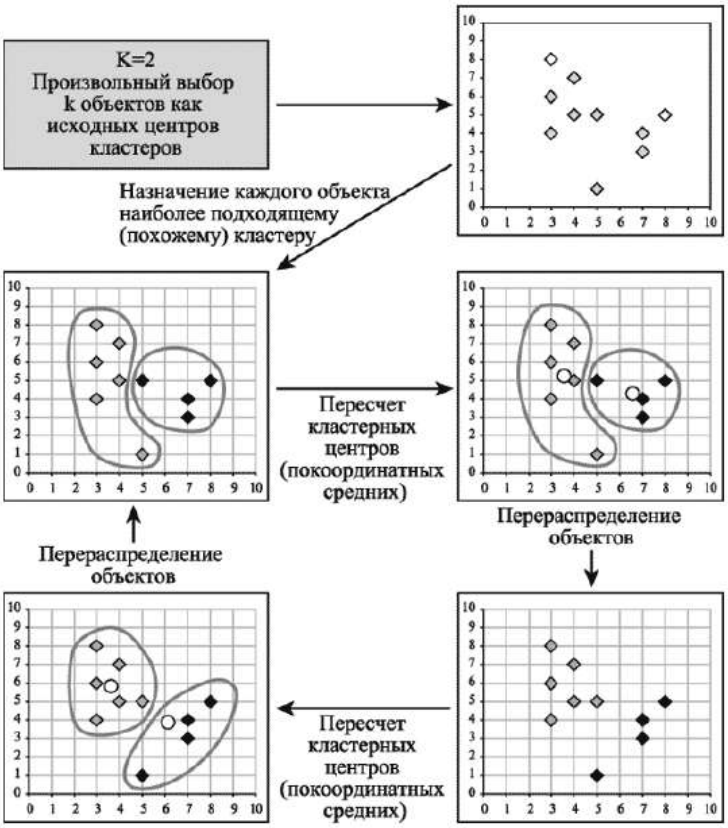


Рисунок - 3. Пример работы алгоритма k-средних (k=2)

Задание и порядок выполнения лабораторной работы №4

Кластерный анализ методом k-средних в R

5. Создать файл с исходными данными.
6. Кластерный анализ проводится в пакете Rcmdr

   Статистика

   Многомерный анализ

   Кластерный анализ

   Кластерный анализ k-средних. В опциях есть возможность выбрать количество кластеров.

   Функция кластерного анализа в R:

```
kmeans(x, centers, iter.шах = 10, nstart = 1, algorithm = c("Hartigan-Wong", "Lloyd", "Forgy", "MacQueen"))
```

Аргументы:

x – численная матрица, содержащая объекты;

centers – или число кластеров, или множество исходных центров кластеров. Если аргумент представляет собой число, то выбирается случайное множество центро кластеров;

iter.шах – максимальное число итераций;

nstart – если centers – число, то данный аргумент определяет, как много случайных множеств может быть выбрано;

algorithm – символ, определяющий используемый алгоритм.

Возвращаемое значение: Объект класса kmeans, который представляет собой список следующих компонент:

cluster – вектор целых чисел, определяющих, в каком кластере размещены объекты;

centers – матрица центров кластеров;

withinss – сумма квадратов расстояний между точками для каждого кластера;

size – число точек в каждом кластере.

7. Провести кластерный анализ экспериментальных данных.
8. Проведя процедуру кластеризации (разбиение на классы или кластеры) несколько раз при различных значениях числа кластеров (от 2-х до 10 кластеров), необходимо выбрать лучшую группировку в смысле критерия минимума отношений средних внутри кластерных и меж кластерных расстояний:

$$F = \frac{d_w / f_w}{d_b / f_b}.$$

Для сравнения нескольких типизаций и выбора наиболее оптимальной из них необходим критерий, численная мера качества классификации.

Одна из оценок качества служит показатель

$$J = J_1 / J_2,$$

где

$$J_1 = \frac{2}{m(m-1)} \sum_{i=1}^{m} \overline{D}_{ii} \quad J_2 = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=i+1}^{m} \overline{D}_{ij} \quad .$$

Здесь $\overline{D}_{ii}$ – среднее расстояние между точками внутри i-го класса, $\overline{D}_{ij}$ – среднее расстояние между парами точек i-го и j-го классов, где m – количество кластеров разбиения.

Полученные результаты оформите в виде Таблицы.

Изобразинm графически значения данного показателя качества классификации. Для этого построить диаграмму, на которой по оси X – количество кластеров, по оси Y – значения показателя J.

Для графической интерпретации используем критерий "каменистой осыпи". Обычно, для выбора размерности какого-либо пространства, используют график зависимости стресса от размерности (график каменистой осыпи). Этот критерий впервые был предложен Кэттелом (Cattell (1966)) в контексте решения задачи снижения размерности в факторном анализе.

Кэттел предложил найти такую абсциссу на графике, в которой график стресса начинает визуально сглаживаться в направлении правой, пологой его части, и, таким образом, уменьшение стресса максимально замедляется. Образно говоря, линия на рисунке напоминает скалистый обрыв, а черные точки на графике напоминают камни, которые ранее упали вниз. Таким образом, внизу наблюдается как бы каменистая осыпь из таких точек. Справа от выбранной точки на оси абсцисс, лежит только "факторная осыпь".

9. Сформулировать выводы.

Пример:

5. Использован файл данных Данные.xls
6. Проведено разбиение на 2 кластера (рис 4)

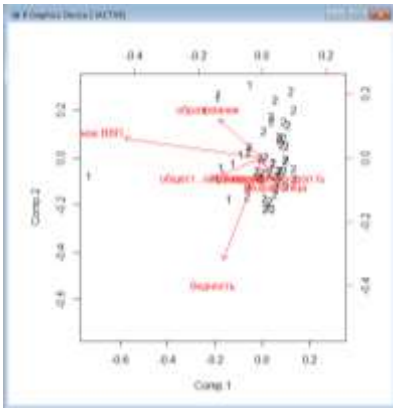


Рисунок 4 – Разбиение данных на 2 кластера

Результаты выполненного анализа:

```
> .cluster$size # Cluster Sizes
[1] 10 44

> .cluster$centers # Cluster Centroids
  new.x.X.ожидаемая.продол.ть new.x.бедность new.x.безработица new.x.образовании
1                   0.6820000      0.7430000         0.8430000         0.9100000
2                   0.6972727      0.7286364         0.8688636         0.8390909
  new.x.общест...ое.развитие new.x.реальное.ВВП.
1                  0.6862000           0.2528000
2                  0.6493182           0.1129318

> .cluster$withinss # Within Cluster Sum of Squares
[1] 0.2077872 0.2390601

> .cluster$tot.withinss # Total Within Sum of Squares
[1] 0.4468473

> .cluster$betweenss # Between Cluster Sum of Squares
[1] 0.2204887
```

– количество элементов в кластере: первый кластер содержит 10 элементов, второй – 44;
– сумма квадратов расстояний внутри кластера: 1 – 0,208, 2 – 0,239
– общая сумма квадратов расстояний внутри кластеров: 0,447
– сумма квадратов расстояний между кластерами – 0,22.

7. Для выбора лучшей группировки в смысле критерия минимума отношений средних внутри кластерных и меж кластерных расстояний было проведено деление на 3 – 10 кластеров и заполнена таблица в MS Excel.

Таблица.

Расчет численного показателя мера качества классификации

Кластеры	D_{ii}	m	D_{ij}	J_1	J_2	J
2	44,01	2	1,43	44,01	0,72	61,47
3	40,03	3	3,83	13,34	1,28	10,47
4	38,64	4	8,62	9,66	2,16	4,48
5	36,36	5	12,07	7,27	2,41	3,01
6	33,21	6	21,03	5,54	3,50	1,58
7	30,43	7	30,84	4,35	4,41	0,99
8	28,66	8	42,84	3,58	5,36	0,67
9	28,11	9	51,65	3,12	5,74	0,54
10	25,41	10	74,03	2,54	7,40	0,34

Графически значения данного показателя качества классификации представлено графически на рис 5. Для этого построена диаграмма, на которой по оси *X* – количество кластеров, по оси *Y* – значения показателя *J*.

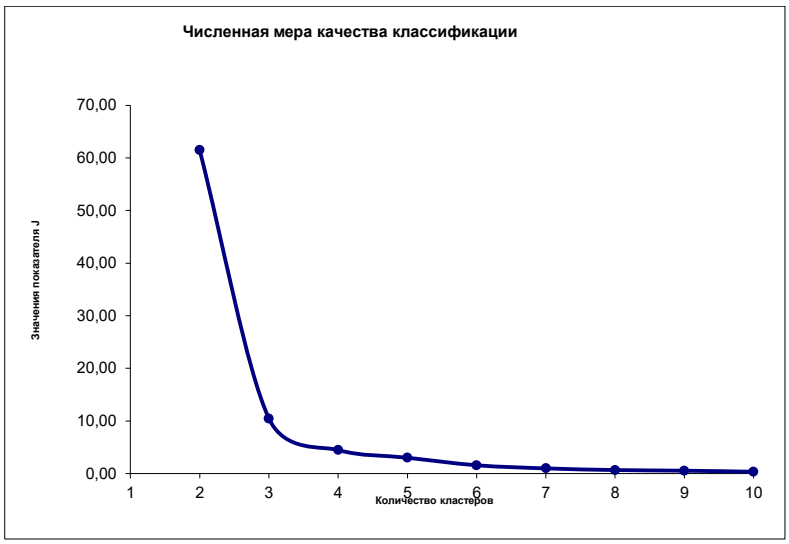


Рисунок – 5. Диаграмма численной меры качества классификации

В соответствии с этим критерием оптимальным разбиением экспериментальных данных является разбиение на 3 кластера.

Иерархический кластерный анализ

1. Провести иерархический кластерный анализ в среде Rcmdr
   Статистика
   Многомерный анализ
   Кластерный анализ
   Иерархический кластерный анализ
2. Провести анализ экспериментальных данных используя разные методы. Полученные результаты сравнить и сделать выводы.

Контрольные вопросы

1. В чем заключается задача кластерного анализа? Для каких задач обработки экспериментальных данных используются методы иерархического кластерного анализа?

2. Перечислите основные меры сравнения объектов между собой. Что такое дендрограмма?

3. Что представляют собой иерархические агломеративные методы кластерного анализа?

4. Что представляют собой иерархические дивизимные методы кластерного анализа?

5. Перечислите основные способы связывания объектов в кластеры. Каким образом определить значимое число кластеров?

6. В чем заключается задача неиерархического кластерного анализа? Для каких задач обработки экспериментальных данных используются методы неиерархического кластерного анализа?

7. В чем суть алгоритма k-средних? Перечислите основные этапы иерархического кластерного анализа по методу k-средних. Каким образом оценить число кластеров в алгоритме k-средних?

Лабораторная работа №5_1
Линейный дискриминантный анализ. Построение канонических и классификационных функций.

Цель:

- Закрепить теоретические знания и приобрести практические навыки в проведении дискриминантного анализа по экспериментальным данным
- исследовать возможности языка R для проведения дискриминантного анализа.

Время: 6 часов

Лабораторное оборудование: персональные компьютеры, выход в сеть Internet, RStudio.

Краткие теоретические сведения

Дискриминантный анализ

С помощью дикриминантного анализа на основании некоторых признаков (независимых переменных) объект может быть причислен к одной из двух или нескольких групп (число групп определяется числом категорий зависимой переменной). В двумерном дискриминантном анализе объекты относятся к одной из двух групп, например, купившие или не купившие данных продукт. А независимыми переменными в этом случае выступают возраст, доход покупателей, и др. показатели.

Дискриминантный анализ используется для анализа данных в том случае, когда зависимая переменная категориальная, а предикторы (независимые переменные) – интервальные. В результате дискриминантного анализа строится так называемая каноническая дискриминантная функция

$$d = b_1 x_1 + b_2 x_2 + \ldots + b_n x_n$$

где x_1 и x_n – значения дискриминантных переменных, соответствующих рассматриваемым случаям, $b_1 \ldots b_n$ – коэффициенты, которые и предстоит оценить с помощью дискриминантного анализа. Коэффициенты подбираются так, чтобы по значениям дискриминантной функции можно было с максимальной четкостью провести разделение по группам.

Процедура дискриминантного анализа состоит из пяти шагов. Первый шаг – формулирование проблемы, требует определения целей, зависимой и независимых переменных. Выборку делят на две части. Анализируемую выборку используют для вычисления дискриминантной функции; проверочную – для проверки достоверности модели. Второй шаг – определение функции, включает выведение такой линейной комбинации предикторов (дискриминантных функций), чтобы группы максимально возможно различались между собой значениями предикторов.

Определение статистической значимости представляет собой третий шаг. Она включает проверку нулевой гипотезы о том, что в совокупности средние всех дискриминантных функций во всех группах равны между собой. Если нулевую гипотезу отклоняют, то имеет смысл интерпретировать результаты.

Четвертый шаг – интерпретация дискриминантных весов или коэффициентов аналогична такой же стадии во множественном регрессионном анализе.

Пятый шаг – проверка достоверности. Она включает разработку классификационной матрицы. Дискриминантные веса, определенные с помощью анализируемой выборки, умножают на значения независимых переменных в проверочной выборке, чтобы получить дискриминантные показатели для случаев в этой выборке. Затем случаи распределяют по группам, исходя из дискриминантных показателей и соответствующего правила принятия решения. Определяют процент верно классифицированных случаев и сравнивают его с процентом случаев, которое можно ожидать на основе классификации методом случайного выбора.

Для оценки коэффициентов существует два известных подхода. Прямой метод включает оценку дискриминантной функции при одновременном введении всех предикторов. Альтернативный ему пошаговый метод включает последовательное введение предсказанных переменных, исходя из их способности дискриминировать группы.

Задание и порядок выполнения лабораторной работы №5_1

Проведение дискриминантного анализа и интерпретация результатов в среде R

Дискриминантный анализ реализован в нескольких пакетах для R, в данной работе будет рассмотрено применение функции lda() из базового пакета MASS.

Все процедуры дискриминантного анализа можно разбить на две группы: первая группа позволяет интерпретировать различия между имеющимися группами (сравнивая средние), вторая – проводить классификацию новых объектов в тех случаях, когда неизвестно заранее, к какому из существующих классов они принадлежат.

1. Подготовка данных для дискриминантного анализа. Для проведения дискриминантного анализ необходимо иметь разделение исходных данных на группы (классы). В данной работе в качестве классов (групп) возьмем разбиение выборки на кластеры.
2. Создать тренировочную выборку из исходных данных с известной группировкой.

Для того чтобы работать с методами классификации с обучением, надо сначала освоить технику «обучения». Для этого выбирается часть данных с известной групповой принадлежностью. На основании анализа этой части (тренировочной выборки) строится гипотеза о том, как должны распределяться по группам остальные, неклассифицированные данные

3. Создать выборку оставшихся данных для последующей проверки классификации
4. Провести дискриминантный анализ по тренировочной выборке используя функцию `lda ()`
5. По полученным данным составить дискриминантную функцию

$$d = b_1 x_1 + b_2 x_2 + \ldots + b_n x_n$$

6. Провести классификацию оставшихся данных и построить матрицу неточностей.
7. По полученным результатам сделать выводы.

Пример. Провести дискриминантный анализ и проверку построенной модели на экспериментальных данных.

1. Подготовка данных для дискриминантного анализа:

Загрузить файл Данные.xls

Провести кластерный анализ методом k-средних на 3 кластера (согласно проведенному анализу в лабораторной работе 4)

Результаты разбиения на кластеры добавить к данным.

2. Создание тренировочной выборки

Создадим выборку строк от 1 до последней с шагом 5

```
Dataset.train <- Dataset [seq (1, nrow(Dataset),5),]
```

Оставшиеся данные сформируем в выборку для последующей проверки полученной классификации

```
Dataset.unknow <- Dataset [-seq (1, nrow(Dataset),5),]
```

3. Проведение дискриминантного анализа.

В качестве переменных используются столбцы 2:7, классификация проводится по столбцу 8:

```
dataset.lda <- lda (Dataset.train[, 2:7], Dataset.train[,8])
```

4. Построение дискриминантных функций:

Для построения дискриминантных функций из проведенного анализ используются Коэффициенты линейных дискриминантов

```
Coefficients of linear discriminants:
                          LD1       LD2
Х.ожидаемая.продол.ть -1060.816 -613.7209
образовании           -1047.838 -677.6436
бедность              -1120.301 -682.0631
безработица           -1079.061 -670.3226
реальное.ВВП.         -1194.556 -675.2506
общест...ое.развитие   5585.588 3383.7527
```

Исходя из результатов получаем две дискриминантные функции

1) $z(x)=5586x_1 - 1195x_2 - 1079x_3 - 1120x_4 - 1048x_5 - 1061x_6$

2) $z(x)=3384x_1 - 675x_2 - 670x_3 - 682x_4 - 678x_5 - 614x_6$

5. Провести классификацию и проверку оставшихся данных:

```
dataset.ldap <- predict(dataset.lda, Dataset.unknow [, 2:7])$class
```

Результат полученной классификации:

```
> dataset.ldap
 [1] 2 2 2 1 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2
[42] 2 2 2 1
Levels: 1 2 3
```

Для проверки созданной модели построим матрицу неточностей:

```
table (dataset.ldap, Dataset.unknow[,8])
```

Результат:

dataset.ldap	1	2
1	3	1
2	6	31
3	0	0

По данной матрице видно, что тренировочная выборка привела к построению гипотезы, по которой в 6 объектов первого класса попали не в «свою» группу и 1 объект из второго класса.

Контрольные вопросы

Лабораторная работа №5_2
Линейный дискриминантный анализ. Проведение дискриминантного анализа и интерпретация результатов.

Цель:

- Закрепить теоретические знания и приобрести практические навыки в проведении дискриминантного анализа по экспериментальным данным
- исследовать возможности языка R для проведения дискриминантного анализа.

Время: 6 часов

Лабораторное оборудование: персональные компьютеры, выход в сеть Internet, RStudio.

Краткие теоретические сведения

Дискриминантный анализ

С помощью дикриминантного анализа на основании некоторых признаков (независимых переменных) объект может быть причислен к одной из двух или нескольких групп (число групп определяется числом категорий зависимой переменной). В двумерном дискриминантном анализе объекты относятся к одной из двух групп, например, купившие или не купившие данных продукт. А независимыми переменными в этом случае выступают возраст, доход покупателей, и др. показатели.

Дискриминантный анализ используется для анализа данных в том случае, когда зависимая переменная категориальная, а предикторы (независимые переменные) – интервальные. В результате дискриминантного анализа строится так называемая каноническая дискриминантная функция

$$d = b_1 x_1 + b_2 x_2 + ... + b_n x_n$$

где x_1 и x_n – значения дискриминантных переменных, соответствующих рассматриваемым случаям, $b_1 ... b_n$ – коэффициенты, которые и предстоит оценить с помощью дискриминантного анализа. Коэффициенты подбираются так, чтобы по значениям дискриминантной функции можно было с максимальной четкостью провести разделение по группам.

Процедура дискриминантного анализа состоит из пяти шагов. Первый шаг – формулирование проблемы, требует определения целей, зависимой и независимых переменных. Выборку делят на две части. Анализируемую выборку используют для вычисления дискриминантной функции; проверочную – для проверки достоверности модели. Второй шаг – определение функции, включает выведение такой линейной комбинации предикторов (дискриминантных функций), чтобы группы максимально возможно различались между собой значениями предикторов.

Определение статистической значимости представляет собой третий шаг. Она включает проверку нулевой гипотезы о том, что в совокупности средние всех дискриминантных функций во всех группах равны между собой. Если нулевую гипотезу отклоняют, то имеет смысл интерпретировать результаты.

Четвертый шаг – интерпретация дискриминантных весов или коэффициентов аналогична такой же стадии во множественном регрессионном анализе.

Пятый шаг – проверка достоверности. Она включает разработку классификационной матрицы. Дискриминантные веса, определенные с помощью анализируемой выборки, умножают на значения независимых переменных в проверочной выборке, чтобы получить дискриминантные показатели для случаев в этой выборке. Затем случаи распределяют по

группам, исходя из дискриминантных показателей и соответствующего правила принятия решения. Определяют процент верно классифицированных случаев и сравнивают его с процентом случаев, которое можно ожидать на основе классификации методом случайного выбора.

Для оценки коэффициентов существует два известных подхода. Прямой метод включает оценку дискриминантной функции при одновременном введении всех предикторов. Альтернативный ему пошаговый метод включает последовательное введение предсказанных переменных, исходя из их способности дискриминировать группы.

Задание и порядок выполнения лабораторной работы №5_2

Проведение дискриминантного анализа и интерпретация результатов в среде R

При проведении дискриминантного анализа может возникнуть вопрос, какие из имеющихся признаков являются информативными при разделении, а какие – сопутствующим балластом.

По построенной модели необходимо выводить важные показатели для оценки ее качества: матрицы неточностей на обучающей выборке, ошибку распознавания и расстояние Махалонобиса между центроидами двух классов

1. Провести шаговую процедуру выбора переменных для построения дискриминантной модели.

Шаговая процедура выбора переменных при классификации, реализованная функцией stepclass() из пакета klaR, основана на вычислении сразу четырех параметров качества моделей-претендентов:

а) индекса ошибок (correctness rate),

б) точности (accuracy), основанной на евклидовых расстояниях между векторами "факта" и "прогноза",

в) способности к разделимости (ability to seperate), также основанной на расстояниях,

г) доверительных интервалах центроидов классов.

```
stepclass(Dataset[,2:7], Dataset[,8], method ="lda")
```

Все эти параметры оцениваются в режиме многократной перекрестной проверки.

2. Построить дискриминантную модель с выбранными переменными, составить уравнение дискриминантной функции.
3. Вывести показатели оценки качества построенной модели: матрица неточностей, ошибку распознавания, расстояние Махалонобиса.

```
table (dataset.ldap, Dataset.unknow[,8])
Err_S <- mean (Dataset.unknow[,8] != dataset.ldap)
mahDist <- dist(dataset.lda$means %*% dataset.lda$scaling)
```

4. Сделать выводы по построенной модели. Сравнить полученные результаты с моделью в которой использовались все переменные.
5. Добавить в выборку данные без классификации, используя дискриминантный анализ провести классификацию.

Контрольные вопросы

1. Категориальные переменные. Приложения дискриминантного анализа. Понятие класса.
2. Дискриминантная функция. Оптимальная процедура классификации.
3. Прямой дискриминантный анализ. Пошаговый дискриминантный анализ.
4. Определение значимости дискриминантной функции. Оценка достоверности дискриминантного анализа. Групповые среднеквадратические отклонения в дискриминантном анализе. Лямбда Уилкса.

Задание и методические указания к расчетно-графической работе

по дисциплине «Интеллектуальный анализ данных»

Тема: Компьютерные методы анализа данных и прогнозирования

Цель: изучить основы методов анализа экспериментальных данных и освоить технику их практического применения в Deduclor Studio

Порядок выполнения

Задание 1. Решение задачи поиска ассоциаций в Deductor
1.1. Подготовить данные для поиска ассоциативных данных.
1.2. Провести поиск ассоциативных правил, проанализировать полученные результаты
Задание 2. Прогнозирование временного ряда.
2.1. Подготовить данные для прогнозирования временного ряда
2.2. Провести анализ временного ряда
2.3. Провести прогнозирование временного ряда на 12 месяцев.

Методические указания к выполнению заданий

1. Решение задачи поиска ассоциаций в Deductor

1.1. Краткие теоретические сведения

Ассоциативные правила позволяют находить закономерности между связанными событиями. Примером такого правила служит утверждение, что покупатель, приобретающий "Хлеб", приобретет и "Молоко". Впервые эта задача была предложена для поиска ассоциативных правил для нахождения типичных шаблонов покупок, совершаемых в супермаркетах, поэтому иногда ее еще называют анализом рыночной корзины (market basket analysis)

Транзакция – это множество событий, произошедших одновременно. Пусть имеется база данных, состоящая из покупательских транзакций. Каждая транзакция – это набор товаров, купленных покупателем за один визит.

Целью анализа является установление следующих зависимостей: если в транзакции встретился некоторый набор элементов X, то на основании этого можно сделать вывод о том, что другой набор элементов Y также должен появиться в этой транзакции. Установление таких зависимостей дает возможность находить очень простые и интуитивно понятные правила.

Основными характеристиками таких правил являются **поддержка** и **достоверность**. Правило "Из *X* следует *Y*" имеет поддержку *s*, если *s*% транзакций из всего набора содержат наборы элементов *X* и *Y*. Достоверность правила показывает, какова вероятность того, что из *X* следует *Y*. Правило "Из *X* следует *Y*" справедливо с достоверностью *c*, если *c*% транзакций из всего множества, содержащих набор элементов *X*, также содержат набор элементов *Y*.

Пример: пусть 75% транзакций, содержащих хлеб, также содержат молоко, а 3% от общего числа всех транзакций содержат оба товара. 75% – это достоверность правила, а 3% –это поддержка.

Лифт – это отношение частоты появления условия в транзакциях, которые также содержат и следствие, к частоте появления следствия в целом. Значения лифта, большие единицы, показывают, что условие появляется более часто в транзакциях, содержащих и следствие, чем в остальных.

Алгоритмы поиска ассоциативных правил предназначены для нахождения всех правил вида "из *X* следует *Y*", причем поддержка и достоверность этих правил должны находиться в рамках некоторых наперед заданных границ, называемых соответственно минимальной и

максимальной поддержкой и минимальной и максимальной достоверностью. Границы значений параметров поддержки и достоверности выбираются таким образом, чтобы ограничить количество найденных правил. Если поддержка имеет большое значение, то алгоритмы будут находить правила, хорошо известные аналитикам или настолько очевидные, что нет никакого смысла проводить такой анализ. С другой стороны, низкое значение поддержки ведет к генерации огромного количества правил, что, конечно, требует существенных вычислительных ресурсов. Тем не менее, большинство интересных правил находится именно при низком значении порога поддержки, хотя слишком низкое значение поддержки ведет к генерации статистически необоснованных правил. Таким образом, необходимо найти компромисс, обеспечивающий, во-первых, интересность правил и, во-вторых, их статистическую обоснованность. Поэтому значения этих границ напрямую зависят от характера анализируемых данных и подбираются индивидуально. Еще одним параметром, ограничивающим Количество найденных правил является максимальная мощность часто встречающихся множеств. Если этот параметр указан, то при поиске правил будут рассматриваться только множества, количество элементов которых будет не больше данного параметра.

Обычные ассоциативные правила – это правила, в которых как в условии, так и в следствии присутствуют только элементы транзакций и при вычислении которых используется только информация о том, присутствует ли элемент в транзакции или нет.

Все множество ассоциативных правил можно разделить на три вида:

Полезные правила – содержат действительную информацию, которая ранее была неизвестна, но имеет логичное объяснение. Такие правила могут быть использованы для принятия решений, приносящих выгоду.

Тривиальные правила – содержат действительную и легко объяснимую информацию, которая уже известна. Такие правила, хотя и объяснимы, но не могут принести какой-либо пользы, т.к. отражают или известные законы в исследуемой области, или результаты прошлой деятельности. При анализе рыночных корзин в правилах с самой высокой поддержкой и достоверностью окажутся товары-лидеры продаж. Практическая ценность таких правил крайне низка.

Непонятные правила – содержат информацию, которая не может быть объяснена. Такие правила могут быть получены или на основе аномальных значений, или глубоко скрытых знаний. Напрямую такие правила нельзя использовать для принятия решений, т.к. их необъяснимость может привести к непредсказуемым результатам. Для лучшего понимания требуется дополнительный анализ.

1.2. Поиск ассоциативных правил в Deductor Studio

Для поиска обычных ассоциативных правил в программе служит обработчик «Ассоциативные правила».

Обработчик требует на входе два поля: идентификатор транзакции и элемент транзакции. Например, идентификатор транзакции – это номер чека или код клиента. А элемент – это наименование товара в чеке или услуга, заказанная клиентом.

Оба поля (идентификатор и элемент транзакции) должны быть дискретного вида.

Затем следует настройка параметров поиска правил. Всего четыре параметра:

Минимальная и максимальная поддержка. Ассоциативные правила ищутся только в некотором множестве всех транзакций. Для того чтобы транзакция вошла в это множество, она должна встретиться в исходной выборке количество раз, больше минимальной поддержки и меньше максимальной. Например, минимальная поддержка равна 1%, а максимальная – 20%. Количество элементов «Хлеб» и «Молоко» столбца «Товар» с одинаковым значением столбца «Номер чека» встречаются в 5% всех транзакций (номеров чека). Тогда эти две строки войдут в искомое множество.

Минимальная и максимальная достоверность. Это процентное отношение количества транзакций, содержащих все элементы, которые входят в правило, к количеству транзакций, содержащих элементы, которые входят в условие. Если транзакция – это заказ, а элемент – товар,

то достоверность характеризует, насколько часто покупаются товары, входящие в следствие, если заказ содержит товары, вошедшие во всё правило.

Пример:

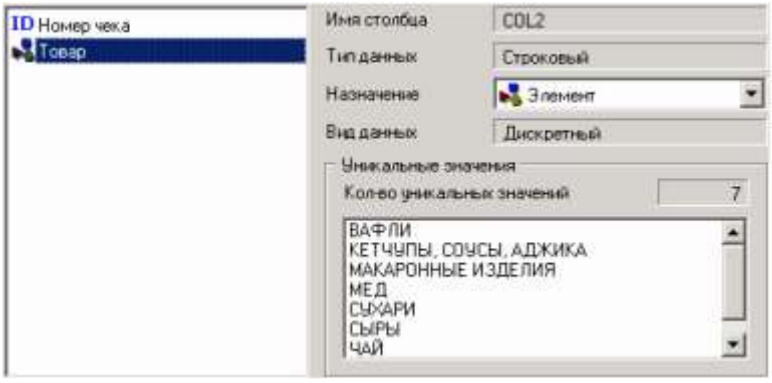
Рассмотрим механизм поиска ассоциативных правил на примере данных о продажах товаров в некоторой торговой точке. Данные представляются в виде таблицы, в которой представлена информация по покупкам продуктов нескольких групп. Она имеет всего два поля "Номер чека" и "Товар".

N	Поле	Тип поля	Назначение
1	ID	строковый	Код чека
2	ITEM	строковый	Товар

Необходимо решить задачу анализа потребительской корзины с целью последующего применения результатов для стимулирования продаж. применения результатов для стимулирования продаж.

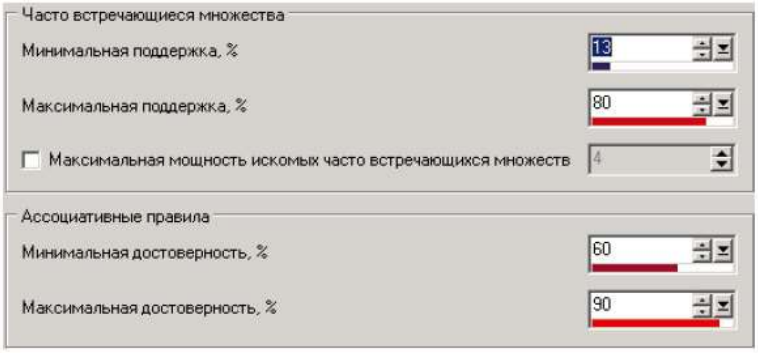
Для поиска ассоциативных правил необходимо запустить **Мастер обработки**. В нем выберать тип обработки "Ассоциативные правила".

На втором шаге Мастера следует указать, какой столбец является идентификатором транзакции (чек), который должен быть дискретным, а какой элементом транзакции (товар).

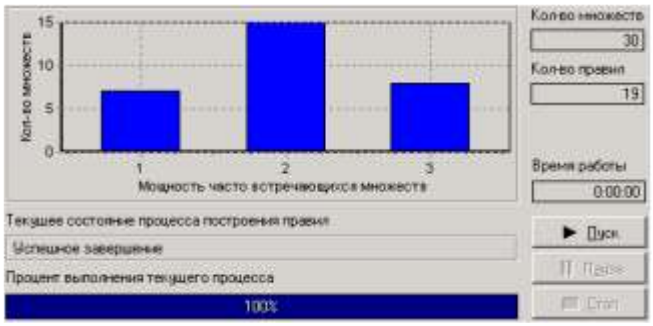


Следующий шаг позволяет настроить параметры построения ассоциативных правил: минимальную и максимальную поддержку, минимальную и максимальную достоверность, а также максимальную мощность множества. Эти параметры необходимо выставлять исходя из характера имеющихся данных. Например, границы поддержки следует указать – 13% и 80% и достоверности 60% и 90%. Можно оставить по умолчанию.

Максимальная мощность искомых часто встречающихся множеств – параметр ограничивает длину k-предметного набора. Например, при установке значения 4 шаг генерации популярных наборов будет остановлен после получения множества 4-предметных наборов. В конечном итоге это позволяет избежать появления длинных ассоциативных правил, которые трудно интерпретируются.



Следующий шаг позволяет запустить процесс поиска ассоциативных правил. На экране отображается информация о количестве множеств и найденных правил, а также числе часто встречающихся множеств.



После завершения процесса поиска полученные результаты можно посмотреть, используя появившиеся специальные визуализаторы "Популярные наборы", "Правила", "Дерево правил", "Что-если".

Популярные наборы - это множества, состоящие из одного и более элементов, которые наиболее часто встречаются в транзакциях одновременно. На сколько часто встречается множество в исходном наборе транзакций, можно судить по поддержке. Данный визуализатор отображает множества в виде списка.

№	Множество	Поддержка	
		%	Кол-во
7	ЧАЙ	75,00	33
3	МАКАРОННЫЕ ИЗДЕЛИЯ	54,55	24
2	КЕТЧУПЫ, СОУСЫ, АДЖИКА	52,27	23
4	МЕД	50,00	22

Получившиеся наборы товаров наиболее часто покупают в данной торговой точке, следовательно можно принимать решения о поставках товаров, их размещении и т.д

Визуализатор "**Правила**" отображает ассоциативные правила в виде списка правил. Этот список представлен таблицей со столбцами: "Номер правила", "Условие", "Следствие", "Поддержка, %", "Поддержка, Количество", "Достоверность".

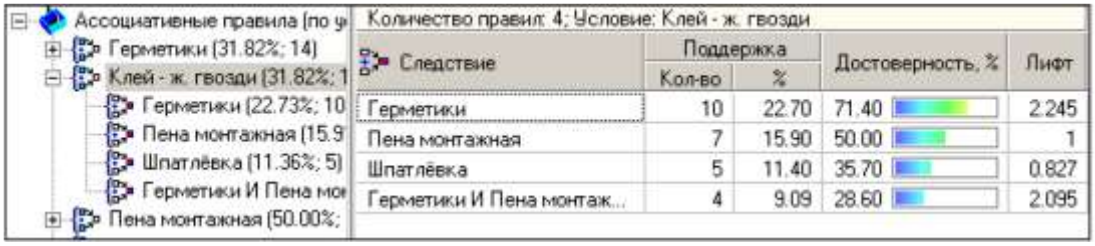
Таким образом, эксперту предоставляется набор правил, которые описывают поведение покупателей.

Правил: 63 из 63 | Фильтр: Без фильтрации

№	Номер правила	Условие	Следствие	Поддержка Кол-во	Поддержка %	Достоверность	Лифт
1	60	Клей - ж. гвозди Шпатлёвка	Герметик Пена монтажная	2	4.55	40.00	2.933
2	57	Герметик Пена монтажная	Клей - ж. гвозди Шпатлёвка	2	4.55	33.33	2.933
3	59	Герметик Шпатлёвка	Клей - ж. гвозди Пена монтажная	2	4.55	40.00	2.514

Например, в правиле с номером 60 в условии присутствуют два элемента: Клей-ж. гвозди и Шпатлёвка. Это правило показывает, что человек, купивший Клей-ж.гвозди и Шпатлёвка с вероятностью 40% купит ещё и Герметики и Пену монтажная.

Визуализатор "**Дерево правил**" – это всегда двухуровневое дерево. Оно может быть построено либо по условию, либо по следствию. При построении дерева правил по условию на первом (верхнем) уровне находятся узлы с условиями, а на втором уровне – узлы со следствием.



Узлы Герметики, Клей-ж. гвозди, Пена монтажная находятся на верхнем уровне дерева и представляют собой условия. А Герметики, Пена монтажная, Шпатлёвка и т.д. – следствия. Это означает, что человек, купивший Клей-ж. гвозди, купит еще и Герметик с достоверностью 71,40%, пену монтажную с достоверностью 50,00% и т.д. В окне слева расположен список со следствиями для конкретного узла с условием. Для каждого следствия указана поддержка, достоверность и лифт. Например, в исходной выборке данных герметики встретились в 10 транзакциях (чеках).

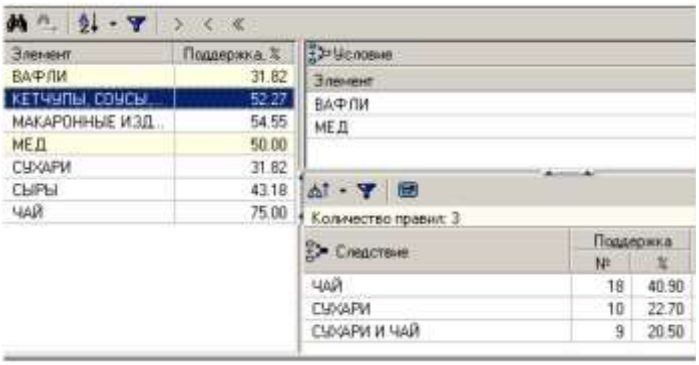
Второй вариант дерева правил – дерево, построенное по следствию. Здесь на первом уровне располагаются узлы со следствием.

Например, для того чтобы человек приобрел Герметик, он должен купить хотя бы один предмет из следующего списка: Клей-ж.гвозди, Пена монтажная, Шпатлёвка и т.д. И для каждого из этих правил отображены поддержка, достоверность и лифт.

Справа от дерева находится список правил, построенный по выбранному узлу дерева. Для каждого правила отображаются поддержка и достоверность. Если дерево построено по условию, то вверху списка отображается условие правила, а список состоит из его следствий. Тогда правила отвечают на вопрос, что будет при таком условии. Если же дерево построено по следствию, то вверху списка отображается следствие правила, а список состоит из его условий. Эти правила отвечают на вопрос, что нужно, чтобы было заданное следствие. Данный визуализатор отображает те же самые правила, что и предыдущий, но в более удобной для анализа форме.

Анализ "Что-если" в ассоциативных правилах позволяет ответить на вопрос, что получим в качестве следствия, если выберем данные условия? Например, какие товары приобретаются совместно с выбранными товарами. В окне слева расположен список всех элементов транзакций. Справа от каждого элемента указана поддержка: сколько раз данный элемент встречается в транзакциях. В правом верхнем углу расположен список элементов, входящих в условие. Это,

например, список товаров, которые приобрел покупатель. Для них нужно найти следствие. Например, товары, приобретаемые совместно с ними. Чтобы предложить человеку то, что он возможно забыл купить. В правом нижнем углу расположен список следствий. Справа от элементов списка отображается поддержка и достоверность. Пусть необходимо проанализировать, что, возможно, забыл покупатель приобрести, если он уже взял вафли и мед. Для этого следует добавить в список условий эти товары (например, с помощью двойного щелчка мыши) и затем нажать на кнопку "Вычислить правила". При этом в списке следствий появятся товары, совместно приобретаемые с данными. В данном случае появятся "Сухари", "Чай", "Сухари и чай", т. е., может быть, покупатель забыл приобрести сухари, чай или и то и другое.



Результаты анализа можно применить и для сегментации покупателей по поведению при покупках, и для анализа предпочтений клиентов, и для планирования расположения товаров в супермаркетах, кросс-маркетинге. Предлагаемый набор визуализаторов позволяет эксперту найти интересные, необычные закономерности, понять, почему так происходит, и применить их на практике.

1.3. Решение задачи поиска ассоциаций в Deductor

Подготовить данные для поиска ассоциативных данных в формате *.txt.

Провести поиск ассоциативных правил, проанализировать полученные результаты

2. Прогнозирование временных рядов

2.1. Краткие теоретические сведения

Временным рядом (рядом динамики, динамическим рядом) называется последовательность значений показателя или признака, упорядоченная в хронологическом порядке, т.е. в порядке возрастания временного параметра. Отдельные наблюдения временного ряда называются **уровнями** этого ряда.

Каждый временной ряд содержит два элемента:

1) значения времени;

2) соответствующие им значения уровней ряда.

В качестве показателя времени в рядах динамики могут указываться либо определенные моменты времени (даты), либо отдельные периоды (сутки, месяцы, кварталы, полугодия, годы и т.д.). В зависимости от характера временного параметра ряды делятся на моментные и интервальные. В моментных рядах динамики уровни характеризуют значения показателя по состоянию на определенные моменты времени. Например, моментными являются временные ряды цен на определенные виды товаров, ряды курсов акций, уровни которых фиксируются для конкретных чисел. Примерами моментных рядов динамики могут служить также ряды численности населения или стоимости основных фондов, т.к. значения уровней этих рядов определяются ежегодно на одно и то же число. В интервальных рядах уровни характеризуют значение показателя за определенные интервалы (периоды) времени. Примерами могут служить

ряды годовой (месячной, квартальной) динамики производства продукции в натуральном или стоимостном выражении.

Если уровни ряда представляют собой не непосредственно наблюдаемые значения, а производные величины (средние или относительные), то такие ряды называются производными. Уровни этих временных рядов получаются с помощью некоторых вычислений на основе абсолютных показателей.

Для успешного изучения динамики процесса важно, чтобы информация была полной, временной ряд имел достаточную длину (с учетом конкретных целей исследования). Например, при изучении периодических колебаний желательно иметь информацию не менее чем за три полных периода колебания. Поэтому при анализе сезонных колебаний на базе рядов месячной или квартальной динамики желательно иметь информацию, как правило, не менее чем за 3 года. Применение определенного математического аппарата также накладывает ограничение на допустимую длину временных рядов. Например, для использования регрессионного анализа требуется иметь временные ряды, длина которых в несколько раз превосходит количество независимых переменных. Уровни рядов динамики могут содержать аномальные значения или «выбросы». Часто появление таких значений может быть вызвано ошибками при сборе, записи и передаче информации. Выявление и исключение таких значений, замена их истинными или расчетными является необходимым этапом первичной обработки данных, т.к. применение математических методов к «засоренной» информации приводит к искажению результатов анализа. Однако аномальные значения могут отражать реальное развитие процесса, как, например, «скачок» курса доллара в «черный вторник».

В практике исследования динамики явлений и прогнозирования принято считать, что значения уровней временных рядов экономических показателей могут содержать следующие компоненты (составные части или структурно-образующие элементы):

- тренд;
- сезонную компоненту;
- циклическую компоненту;
- случайную составляющую.

Под **трендом** понимают изменение, определяющее общее направление развития, основную тенденцию временного ряда. Это систематическая составляющая долговременного действия. Наряду с долговременными тенденциями во временных рядах экономических процессов часто имеют место более или менее регулярные колебания – периодические составляющие рядов динамики. Если период колебаний не превышает одного года, то их называют **сезонными**. Чаще всего причиной их возникновения считаются природно-климатические условия. Примером могут служить колебания цен на сельскохозяйственную продукцию, в частности на картофель.

При большем периоде колебания считают, что во временных рядах имеет место **циклическая** составляющая. Примерами могут служить демографические, инвестиционные и другие циклы. Если из временного ряда удалить тренд и периодические составляющие, то останется **нерегулярная** компонента.

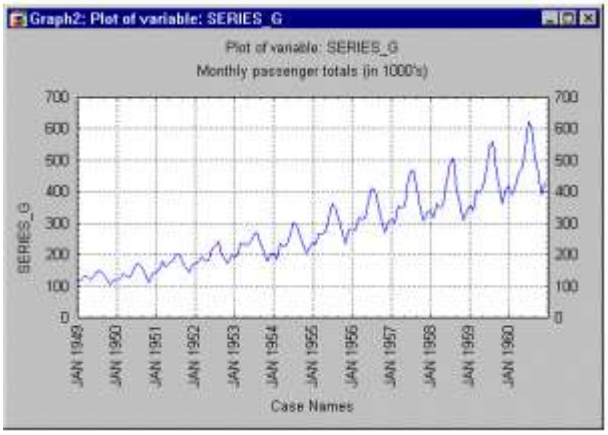
Разделяют факторы, под действием которых формируется нерегулярная компонента, на 2 вида:

- факторы резкого, внезапного действия;
- текущие факторы.

Факторы первого вида (например, стихийные бедствия, эпидемии и др.), как правило, вызывают более значительные отклонения. Иногда такие отклонения называют катастрофическими колебаниями. Факторы второго вида вызывают случайные колебания, являющиеся результатом действия большого числа побочных причин. Влияние каждого из текущих факторов незначительно, но ощущается их суммарное воздействие.

Пример. Ряд G представляет месячные международные авиаперевозки (в тысячах) в течение 12 лет с 1949 по 1960. График месячных перевозок ясно показывает почти линейный

тренд, т.е. имеется устойчивый рост перевозок из года в год (примерно в 4 раза больше пассажиров перевезено в 1960 году, чем в 1949).



В то же время характер месячных перевозок повторяется, они имеют почти один и тот же характер в каждом годовом периоде (например, перевозок больше в отпускные периоды, чем в другие месяцы). Этот пример показывает довольно определенный тип модели временного ряда, в которой амплитуда сезонных изменений увеличивается вместе с трендом.

2.2. Прогнозирование в Deductor Studio

Прогнозирование результата на определенное время вперед, основываясь на данных за прошедшее время, – задача, встречающаяся довольно часто. К примеру, перед большинством торговых фирм стоит задача оптимизации складских запасов, для решения которой требуется знать, что и сколько должно быть продано через неделю и т.п., задача предсказания стоимости акций какого-нибудь предприятия через день и т.д. и другие подобные вопросы. Deductor Studio предлагает для этого инструмент "Прогнозирование".

Прогнозирование появляется в списке Мастера обработки только после построения какой-либо модели прогноза: нейросети, линейной регрессии и т.д. Прогнозировать на несколько шагов вперед имеет смысл только временной ряд (к примеру, если есть данные по недельным суммам продаж за определенный период, можно спрогнозировать сумму продаж на две недели вперед).

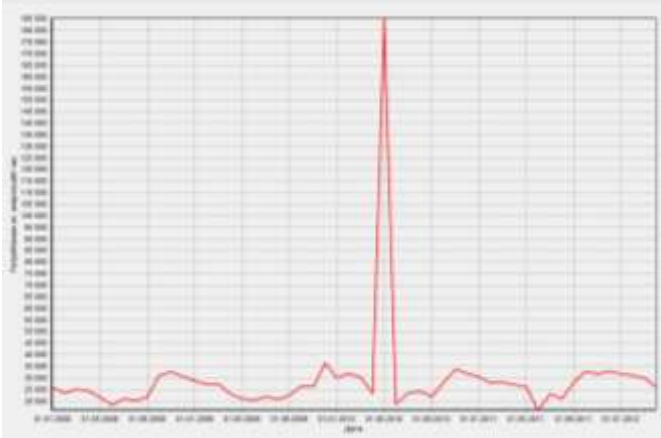
Пример. Пример исходных данных по потреблению электроэнергии находятся в Табл.

Таблица - Пример данных для прогнозирования временного ряда

Дата	Потребление эл. энергия,кВт час
01.01.2008	25668
01.02.2008	23292
01.03.2008	25155
01.04.2008	24228
01.05.2008	21510
01.06.2008	18513
01.07.2008	20960,95
01.08.2008	20079
01.09.2008	21951
01.10.2008	31212
01.11.2008	32688
01.12.2008	30438
01.01.2009	28764

Дата	Потребление эл. энергия,кВт час
01.02.2009	27387
01.03.2009	27171
01.04.2009	23037,669
01.05.2009	20960,95
01.06.2009	20361,525
01.07.2009	21972,3
01.08.2009	20659,725
01.09.2009	22602,975
01.10.2009	26337,15
01.11.2009	26279,25
01.12.2009	36532,425
01.01.2010	29880,45
01.02.2010	32113,125
01.03.2010	30285,675
01.04.2010	23484,975
01.05.2010	185127,5
01.06.2010	18588,675
01.07.2010	23471,475
01.08.2010	24175,5
01.09.2010	21903,825
01.10.2010	28014,15
01.11.2010	33909
01.12.2010	32032,35
01.01.2011	30678,75
01.02.2011	27801
01.03.2011	28182,225
01.04.2011	26865,375
01.05.2011	26390,55
01.06.2011	15997,5
01.07.2011	23166,45
01.08.2011	20960,95
01.09.2011	27836,175
01.10.2011	32928,75
01.11.2011	31778,85
01.12.2011	32908,425
01.01.2012	31761,3
01.02.2012	31200,45
01.03.2012	29871,3
01.04.2012	25868,4

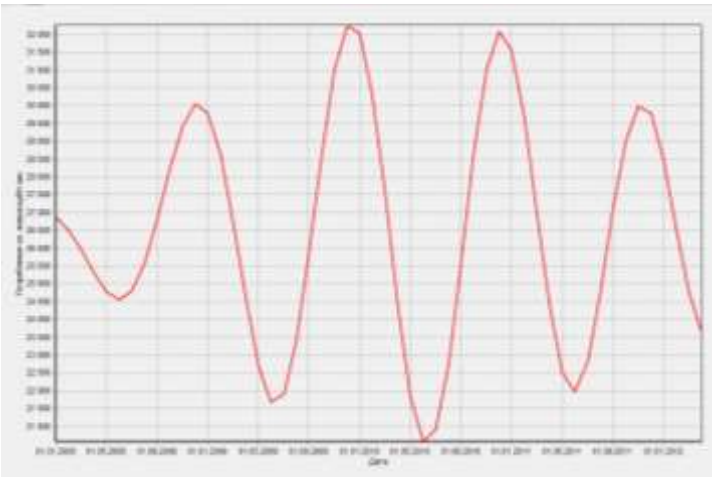
После импорта данных воспользуемся диаграммой для их просмотра.



На ней видно, что данные содержат аномалии (выбросы) и шумы, за которыми трудно разглядеть тенденцию. Поэтому перед прогнозированием необходимо удалить аномалии и сгладить данные.

Следующим шагом необходимо провести «Редактирование выбросов и экстремальных значений» и «Спектральную обработку»

Для просмотра данных можно воспользоваться диаграммой. Видно, что данные сгладились, аномалии и шумы исчезли.



Строить прогноз на будущее будем, основываясь на данных прошлых периодов, т. е. предполагая, что количество потребляемой энергии на следующий месяц зависит от количества потребляемой энергии за предыдущие месяцы. Это значит, что входными факторами для модели могут быть количество потребляемой энергии за текущий месяц, продажи за месяц ранее и т.д., а результатом должно быть количество потребляемой энергии за следующий месяц.

Для дальнейшей работы необходимо трансформировать данные к скользящему окну.

Скользящее окно

При решении некоторых задач, например, при прогнозировании временных рядов при помощи нейросети, требуется подавать на вход модели значения нескольких смежных отсчетов из исходного набора данных. Такой метод отбора данных называется скользящим окном (окно – поскольку выделяется только некоторый непрерывный участок данных, скользящее – поскольку это окно «перемещается» по всему набору). При этом эффективность реализации заметно повышается, если не выбирать данные каждый раз из нескольких последовательных записей, а последовательно расположить данные, относящиеся к конкретной позиции окна, в одной записи.

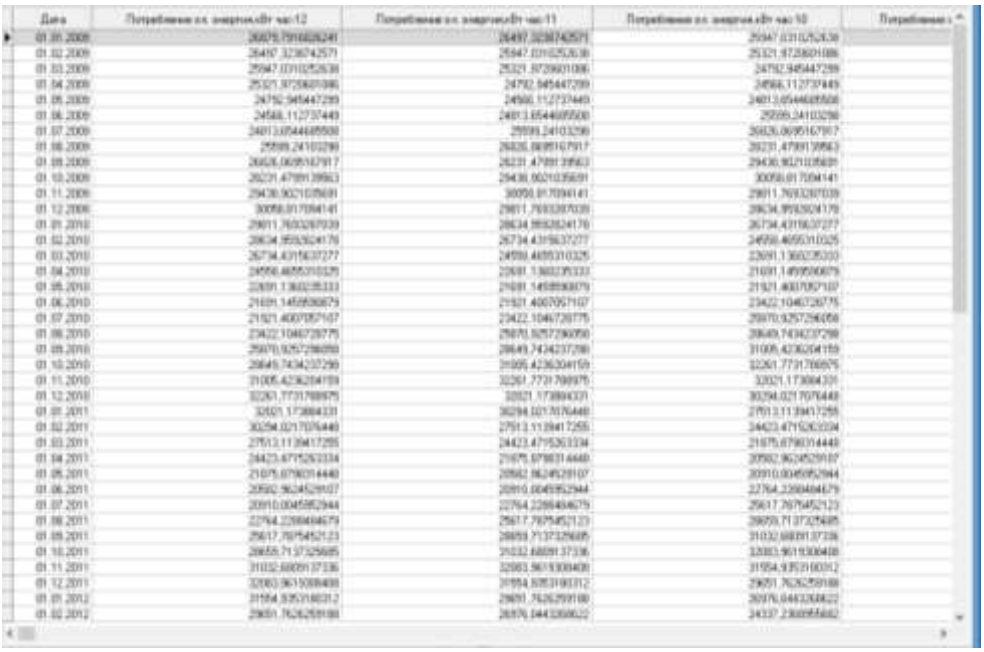
Значения в одном из полей записи будут относиться к текущему отсчету, а в других – смещены от текущего отсчета «в будущее» или «в прошлое». Таким образом, преобразование скользящего окна имеет два параметра: «глубина погружения» - количество «прошлых»

отсчетов, попадающих в окно, и «горизонт прогнозирования» – количество «будущих» отсчетов. Следует отметить, что для граничных (относительно начала и конца всей выборки) положений окна будут формироваться неполные записи, т.е. записи, содержащие пустые значения для отсутствующих прошлых или будущих отсчетов. Алгоритм преобразования позволяет исключить такие записи из выборки (тогда для нескольких граничных отсчетов записи формироваться не будут) либо включить их (тогда формируются записи для всех имеющихся отсчетов, но некоторые из них будут неполными). Для правильного формирования скользящего окна данные должны быть соответствующим образом упорядочены.

Скользящее окно 12 месяцев назад

Запустить Мастер обработки, выбрать в качестве обработчика скользящее окно и перейти на следующий шаг.

Требуется выбрать глубину погружения 12, назначив поле "Количество" используемым. Тогда данные трансформируются к скользящему окну так, что аналитику будут доступны все нужные факторы для построения прогноза.



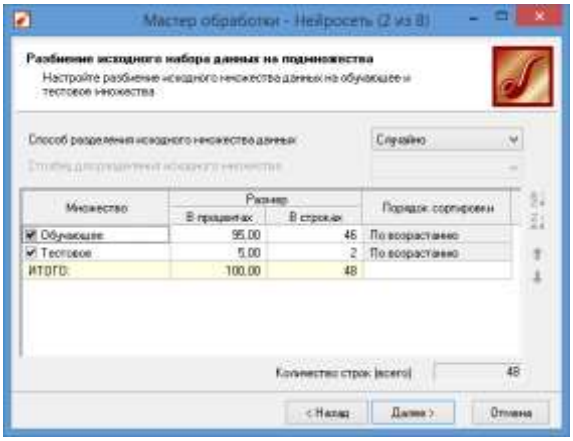
Теперь в качестве входных факторов можно использовать "Потребление эл. энергии - 12", " Потребление эл. энергии - 11" - данные по количеству потребляемой энергии 12 и 11 месяцев назад (относительно прогнозируемого месяца), и остальные. В качестве выходного поля можно использовать столбец " Потребление эл. энергии".

Обучение нейросети

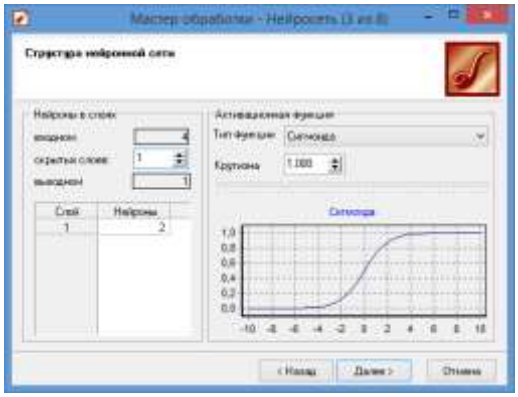
Построение модели прогноза.

Открыть Мастер обработки и выбрать в нем нейронную сеть. На втором шаге Мастера необходимо установить входные поля. Например можно взять "Потребление эл. энергии - 4", "Потребление эл. энергии - 3" "Потребление эл. энергии - 2" и "Потребление эл. энергии - 1"а в качестве выходного - " Потребление эл. энергии ". Остальные поля сделать информационными.

На следующем шаге необходимо указать разбиение тестового и обучающего множеств.



На следующем этапе отмечаются необходимое количество слоев и нейронов в нейросети.

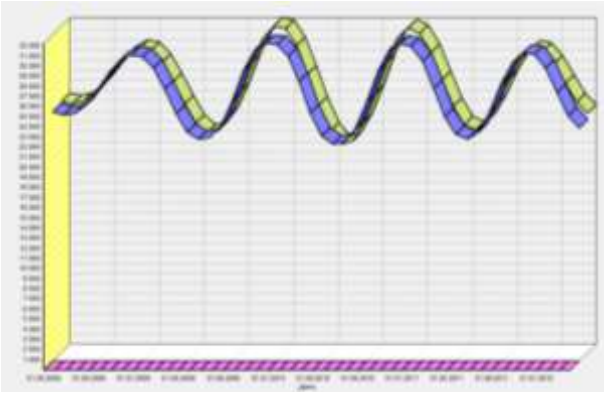


Перейдя далее, выберается алгоритм обучения нейросети.

После построения модели для просмотра качества обучения полученные данные представить в виде диаграммы и диаграммы рассеяния.

В Мастере настройки диаграммы выбрать для отображения поля " Потребление эл. энергии " и " Потребление эл. энергии _OUT" - реальное и спрогнозированное значение.

Результатом будет два графика.

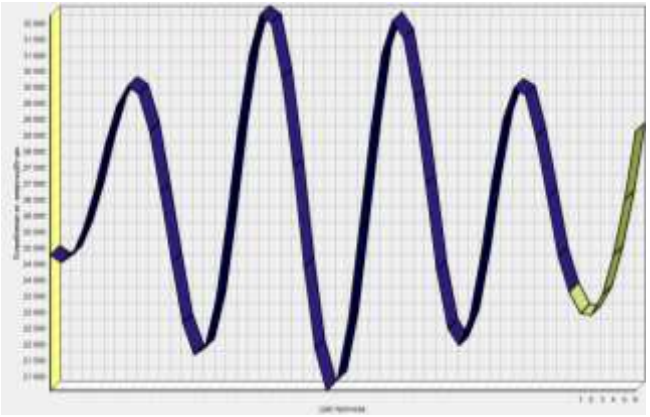


Построение прогноза

Нейросеть обучена, осталось получить требуемый прогноз. Для этого неоходимо открыть Мастер обработки и выбирать появившийся теперь обработчик "Прогнозирование".

На втором шаге Мастера предлагается настроить связи столбцов для прогнозирования временного ряда: откуда брать данные для столбца при очередном шаге прогноза.

Мастер сам верно настроил все переходы, поэтому остается только указать горизонт прогноза (на сколько вперед будем прогнозировать), а также для наглядности следует добавить к прогнозу исходные данные, установив в Мастере соответствующийфлажок.



Результат

После этого необходимо в качестве визуализатора выбрать "Диаграмму прогноза", которая появляется только после прогнозирования временного ряда.

В Мастере настройки столбцов диаграммы прогноза надо указать в качестве отображаемого столбец "Количество", а в качестве подписей по оси X указать столбец "Шаг прогноза".

Теперь аналитик может узнать количество потребленной электроэнергии в следующем месяце и шесть месяцев спустя.

2.3 Решение задачи прогнозирования временного ряда в Deductor

Подготовить данные для прогнозирования временного ряда в формате *.txt
Провести анализ временного ряда
Провести прогнозирование временного ряда на 12 месяцев.
Сделать выводы

Контрольные вопросы

1. Стохастический процесс. Стационарный стохастический процесс в слабом (широком) смысле. Нормальный стохастический процесс. Белый шум.
2. Параметры стационарного процесса
3. Методы для распознавания стационарности временных рядов.
4. Примеры параметрических тестов проверки временных рядов на стационарность.
4. Процессы авторегрессии.
5. Процессы скользящего среднего.
6. Процессы авторегрессии - скользящего среднего.
7. Автоковариационная функция. Автокорреляционная функция для идентификации модели стационарного стохастического процесса. Частная автокорреляционная функция для идентификации модели стационарного стохастического процесса.

Требования к содержанию и оформлению отчетов

Отчеты по лабораторным работам оформляются согласно правилам оформления принятым на кафедре, ГОСТам и ЕСКД.

Основные правила по оформлению отчетной документации:

Параметры страницы: А4 (21×29,7), ориентация – книжная (допускается использовать альбомную ориентацию страницы для выполнения схем и таблиц).

Поля: левое – 25 (30) мм, верхнее – 20 мм, нижнее – 20 мм, правое – 10 мм.

Нумерация страницы – вверху, по центру. Нумерация ведется с титульного листа, номер на титульном листе не ставиться.

Шрифт Times New Roman, кегль 14, интервал – полуторный.

Заголовки разделов: абзацный отступ – 0, выравнивание по центру, шрифт – жирный, буквы прописные, нумерация – арабскими цифрами, точка в конце номера и названия раздела не ставиться.

Заголовки подразделов (допускается три уровня, например, 1.1., 1.1.1.): абзацный отступ – 1,25 см, выравнивание по ширине, шрифт – жирный, точка в конце названия подраздела не ставиться.

Основной текст: абзацный отступ – 1,25 см, выравнивание по ширине, шрифт – обычный.

Нумерация рисунков и таблиц – сквозная внутри раздела (например, в разделе 1 – рисунок 1.1, рисунок 1.2 и т.д., или таблица 1.1, таблица 1.2 и т.д.).

Рисунки помещаются после упоминания их в тексте и имеют подпись, размещаемую под рисунком без абзацного отступа и имеющую выравнивание по центру и точку на конце названия (например, Рисунок 1.1 – Название.).

Таблицы размещаются после ссылки на них в тексте. Название приводится над таблицей, без абзацного отступа с выравниванием по левому краю, без точки на конце названия (например, (Таблица 2.2 – Название).

Допускается выносить рисунки и таблицы в Приложения. В этом случае ссылка должна содержать номер приложения (например: рисунок А.1 Приложения А или таблица Б.1 Приложения Б).

Основная часть должна содержать ссылки на используемую литературу или информационные источники, список которых приводится после раздела Выводы и перед Приложениями. Ссылка заключается в квадратные скобки (например – [1], [5,7], [3-6].

Приложения обозначаются русскими заглавными буквами в порядке их следования (Приложение А, Приложение Б). Слово «Приложение…» выравнивается по центру без абзацного отступа и имеет жирный шрифт, прописные буквы. Название приложение располагается на следующей строке, без абзацного отступа, выравнивание по центру, шрифт – жирный, первая буква прописная, остальные – строчные.

По завершению изучения курса у студента должен быть сформировать набор отчетов к лабораторным работам (Приложение №1), сведенных в единый документ и имеющий единый титульный лист (Приложение №2), отчет к расчетно-графической работе (Приложение №3) на котором отражаются результаты прохождения этапов изучения дисциплины.

Каждый раздел этого документа является отчетом по выполнению соответствующей лабораторной работы (обязательные разделы и правила выполнения отчетов представлены в Приложении 1).

Сформированный документ, с отметками о выполнении всех лабораторных работ обязателен для представления на итоговом контроле и является подтверждением о допуске к итоговому контролю.

К отчету прилагается папка с файлами – результатами выполнения лабораторной работы (данная папка должна так же находится на сетевом диске в папке проектов изучаемой дисциплины), название папки ИАД_фамилия.

Организация защиты и критерии оценивания выполнения лабораторных работ

К защите представляется отчет, включающий в себя результаты выполнения лабораторной работы, выполненный согласно правилам и единый титульный лист, на котором отмечаются результаты выполнения заданий.

К отчетам прилагается электронный носитель, содержащий папки с исполняемыми файлами, файлами отчетов и презентациями (если требуется в задании) созданных в ходе выполнения лабораторных работ.

На проверку теоретической подготовки, проводимой по контрольным вопросам, отводиться 5–6 минут.

Степень усвоения теоретического материала оценивается по следующим критериям:

- ***оценка «отлично» выставляется, если:***
  - последовательно, четко, связно, обоснованно и безошибочно с использованием принятой терминологии изложен учебный материал, выделены главные положения, ответ подтвержден конкретными примерами, фактами;
  - самостоятельно и аргументировано сделан анализ, обобщение, выводы, установлены межпредметные (на основе ранее приобретенных знаний) и внутрипредметные связи, творчески применены полученные знания в незнакомой ситуации;
  - самостоятельно и рационально используются справочные материалы, учебники, дополнительная литература, первоисточники; применяется систему условных обозначений при ведении записей, сопровождающих ответ; используются для доказательства выводы из наблюдений и опытов, ответ подтверждается конкретными примерами;
  - допускает не более одного недочета, который легко исправляется по требованию преподавателя.
- ***оценка «хорошо» ставится, если:***
  - дан полный и правильный ответ на основе изученных теорий; допущены незначительные ошибки и недочеты при воспроизведении изученного материала, определения понятий, неточности при использовании научных терминов или в выводах и обобщениях из наблюдений и опытов; материал излагает в определенной логической последовательности;
  - самостоятельно выделены главные положения в изученном материале; на основании фактов и примеров проведено обобщение, сделаны выводы, установлены внутрипредметные связи.
  - допущены одна негрубая ошибку или не более двух недочетов, которые исправлены самостоятельно при требовании или при небольшой помощи преподавателя; в основном усвоил учебный материал.
- ***оценка «удовлетворительно» ставится, если:***
  - усвоено основное содержание учебного материала, но имеются пробелы в усвоении материала, не препятствующие дальнейшему изучению; материал излагает несистематизированно, фрагментарно, не всегда последовательно;
  - показана недостаточная сформированность отдельных знаний и умений; выводы и обобщения аргументируются слабо, в них допускаются ошибки;
  - допущены ошибки и неточности в использовании научной терминологии, даются недостаточно четкие определения понятий; в качестве доказательства не используются выводы и обобщения из наблюдений, фактов, опытов или допущены ошибки при их изложении;
  - обнаруживается недостаточное понимание отдельных положений при воспроизведении текста учебника (записей, первоисточников) или неполные ответы на вопросы преподавателя, с допущением одной – двух грубых ошибок.

- ***оценка «неудовлетворительно» ставится, если:***
  - не усвоено и не раскрыто основное содержание материала; не сделаны выводы и обобщения;
  - не показано знание и понимание значительной или основной части изученного материала в пределах поставленных вопросов или показаны слабо сформированные и неполные знания и неумение применять их к решению конкретных вопросов и задач по образцу;
  - при ответе (на один вопрос) допускается более двух грубых ошибок, которые не могут быть исправлены даже при помощи преподавателя;
  - не даются ответы ни на один их поставленных вопросов.

Оценка выполнения лабораторных работ проводится по следующим критериям

- ***оценка «отлично» ставится, если студент:***
  - творчески планирует выполнение работы;
  - самостоятельно и полностью использует знания программного материала;
  - правильно и аккуратно выполняет задание;
  - умеет пользоваться литературой и различными информационными источниками;
  - выполнил работу без ошибок и недочетов или допустил не более одного недочета
- ***оценка «хорошо» ставится, если студент:***
  - правильно планирует выполнение работы;
  - самостоятельно использует знания программного материала;
  - в основном правильно и аккуратно выполняет задание;
  - умеет пользоваться литературой и различными информационными источниками;
  - выполнил работу полностью, но допустил в ней: не более одной негрубой ошибки и одного недочета или не более двух недочетов.
- ***оценка «удовлетворительно» ставится, если студент:***
  - допускает ошибки при планировании выполнения работы;
  - не может самостоятельно использовать значительную часть знаний программного материала;
  - допускает ошибки и неаккуратно выполняет задание;
  - затрудняется самостоятельно использовать литературу и информационные источники;
  - правильно выполнил не менее половины работы или допустил:
    - не более двух грубых ошибок или не более одной грубой и одной негрубой ошибки и одного недочета;
    - не более двух– трех негрубых ошибок или одной негрубой ошибки и трех недочетов;
    - при отсутствии ошибок, но при наличии четырех–пяти недочетов.
- ***оценка «неудовлетворительно» ставится, если студент:***
  - не может правильно спланировать выполнение работы;
  - не может использовать знания программного материала;
  - допускает грубые ошибки и неаккуратно выполняет задание;
  - не может самостоятельно использовать литературу и информационные источники;
  - допустил число ошибок недочетов, превышающее норму, при которой может быть выставлена оценка «3»;
  - если правильно выполнил менее половины работы;
  - не приступил к выполнению работы;
  - правильно выполнил не более 10% всех заданий.

Список литературы

Основная литература

1. Агалаков, С.А. Статистические методы анализа данных [Электронный ресурс]: учебное пособие / С.А. Агалаков. – Электрон. дан. – Омск: ОмГУ, 2017. – 92 с. – Режим доступа: https://e.lanbook.com/book/103047
2. Замятин, А.В. Интеллектуальный анализ данных [Электронный ресурс]: учебное пособие / А.В. Замятин. – Электрон. дан. – Томск : ТГУ, 2016. – 120 с. – Режим доступа: https://e.lanbook.com/book/74565
3. Интеллектуальный анализ данных средствами MS SQL Server 2008 [Электронный ресурс]: учебное пособие. – Электрон. дан. – Москва, 2016. – 337 с. – Режим доступа: https://e.lanbook.com/book/100609.

Дополнительная литература

4. Базовые и прикладные информационные технологии: Учебник / Гвоздева В. А. – М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2015. – 384 с.: 60x90 1/16. – (Высшее образование) (Переплёт 7БЦ) ISBN 978-5-8199-0572-2. Режим доступа: http://znanium.com/catalog.php?bookinfo=504788
5. Статистический анализ данных в MS Excel : учеб. пособие / А.Ю. Козлов, В.С. Мхитарян, В.Ф. Шишов. – М. : ИНФРА-М, 2017. – 320 с. – (Высшее образование: Бакалавриат). — www.dx.doi.org/10.12737/2842. - Режим доступа: http://znanium.com/catalog/product/858510

Информационные ресурсы, необходимые для освоения дисциплины

№	Адрес сайта и его описание	Перечень материалов, представленных на сайте
1.	http://e.lanbook.com	Электронная библиотечная система «Издательства «Лань»»
2.	http://znanium.com	Электронно-библиотечная система Znanium.com
3.	http://machinelearning.ru.	Профессиональный информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных
4.	https://www.r-project.org	Проект R для статистических вычислений
5.	https://basegroup.ru/deductor/manual	Платформа Deductor. Руководство

Приложение 1

Образец оформления и содержания отчета по лабораторной работе

Лабораторная работа №___

Тема:

Цель:

1. Краткие теоретические сведения по изучаемой теме

…

2. Отчет о выполнении задания (согласно плану, представленному в методических указаниях)

…

Выводы

…

Список литературы и информационных источников

…

Приложения

Приложение 2

Образец единого титульного листа к отчетам по лабораторным работам

Министерство образования и науки Российской Федерации
ФГАОУ ВО «Севастопольский государственный университет»

Институт информационных технологий и управления в технических системах

Кафедра «Информационные системы»

Сводный отчет по лабораторному практикуму
по дисциплине «Интеллектуальный анализ данных»

№ п/п	Оценка выполнения				Подпись
	Теория	Практика	Итог	Дата	
1					
2					
3					
4					
5					

Выполнил: студент(ка) группы ___
ФИО

Принял: должность ФИО

г.Севастополь
20__ г.

Приложение 3

Образец титульного листа к отчету по расчетно-графической работе

Министерство образования и науки Российской Федерации
ФГАОУ ВО «Севастопольский государственный университет»

Институт информационных технологий и управления в технических системах

Кафедра «Информационные системы»

Расчетно-графическая работа
по дисциплине «Интеллектуальный анализ данных»

Тема: Компьютерные методы анализа данных и прогнозирования

Вариант № __

Выполнил: студент(ка) группы ___
ФИО

Принял: должность ФИО

г.Севастополь
20__ г.