

Министерство образования и науки Украины
Севастопольский национальный технический
университет



МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к практическим занятиям по дисциплине
«Автоматизированные системы управления предприятием».

Часть 1. «Интеллектуальный анализ данных с использованием
языка R»

для студентов направления подготовки 8.091501 —
«Компьютерные системы и сети» дневной формы обучения

Севастополь – 2014

Методические указания к практическим занятиям по дисциплине «Автоматизированные системы управления предприятием». Часть 1. «Интеллектуальный анализ данных с использованием языка R» для студентов направления подготовки 8.091501 — «Компьютерные системы и сети» дневной формы обучения / Сост.: Е.Н. Машенко, О.В. Новикова. — Севастополь: Изд-во СевНТУ, 2014. — 37 с.

Целью методических указаний является оказание помощи студентам в выполнении практических занятий. Приводятся постановка задачи на практическое занятие, варианты заданий, изложены краткие теоритические сведения и примеры программ.

Методические указания рассмотрены и утверждены на заседании кафедры кибернетики и вычислительной техники (протокол № 11 от 25 июня 2014 г.)

Допущено учебно-методическим центром СевНТУ в качестве методических указаний.

Рецензент: Шевченко В.И., старший преподаватель
кафедры кибернетики и вычислительной техники

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №1. ИНСТРУМЕНТЫ DATA MINING. ИНСТАЛЛЯЦИЯ ЯЗЫКА R.....	5
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 2. ОСНОВЫ РАБОТЫ СО СРЕДОЙ R.....	6
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 3. МЕТОДЫ ВИЗУАЛИЗАЦИИ В СРЕДЕ R.....	10
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 4. КЛАССИФИКАЦИЯ.....	18
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 5. КЛАСТЕРИЗАЦИЯ.....	20
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	24
ПРИЛОЖЕНИЕ А - Установка среды разработки на R для Windows 7	25
ПРИЛОЖЕНИЕ Б - Установка R в Ubuntu 13.04 Raring Ringtail.....	31
ПРИЛОЖЕНИЕ В – Пример проверочной программы для лабораторной работы №1	33
ПРИЛОЖЕНИЕ Г – Установка нового пакета для среды разработки....	34
ПРИЛОЖЕНИЕ Д – Справочная информация по некоторым базовым функциям языка программирования R.....	36

ВВЕДЕНИЕ

Информация стала одним из важнейших стратегических, управленческих ресурсов, наряду с ресурсами – человеческим, финансовым, материальным. Ее производство и потребление составляют необходимую основу эффективного функционирования и развития различных сфер общественной жизни, и, прежде всего, экономики. А это означает, что не только каждому человеку становятся доступными источники информации в любой части нашей планеты, но и генерируемая им новая информация становится достоянием всего человечества. В современных условиях право на информацию и доступ к ней имеет жизненную ценность для всех членов общества. Возрастающая роль информации в обществе явилась предметом научного осмысления. Были выдвинуты теории, объясняющие ее место и значение. Наиболее популярными являются теории постиндустриального и информационного общества [1].

Кластерный анализ позволяет открыть в данных ранее неизвестные закономерности, которые практически невозможно исследовать другими способами и представить их в удобной для пользователя форме. Использование методов кластерного анализа предполагает выделение компактных, удаленных друг от друга групп объектов, отыскивает «естественное» разбиение совокупности на области скопления объектов. Обычно исходные данные представлены в виде матриц близости или расстояний между объектами либо в виде точек в многомерном пространстве. Наиболее распространены данные второго вида, для которых кластерный анализ ориентирован на выделение некоторых геометрически удаленных групп, внутри которых объекты близки.

Спектр применений кластерного анализа очень широк. Он используется в археологии, медицине, психологии, химии, биологии, государственном управлении, маркетинге, социологии и других дисциплинах. Таким образом, возрастает необходимость грамотного владения инструментами кластерного анализа и средствами визуализации полученных результатов.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №1. ИНСТРУМЕНТЫ DATA MINING. ИНСТАЛЛЯЦИЯ ЯЗЫКА R.

Цель работы: установить среду разработки для языка R и протестировать её работоспособность.

Теоретические сведения.

R — язык программирования для статистической обработки данных и работы с графикой, а также свободная программная среда вычислений с открытым исходным кодом в рамках проекта GNU. Язык создавался как аналогичный языку S, разработанному в Bell Labs и является его альтернативной реализацией, хотя между языками есть существенные отличия, но в большинстве своём код на языке S работает в среде R. Изначально R был разработан сотрудниками статистического факультета Оклендского университета Россом Айхэкой и Робертом Джентлменом (первая буква их имён — R), на момент 2011 года язык и среда поддерживаются и развиваются организацией R Foundation.

R широко используется как статистическое программное обеспечение для анализа данных и фактически стал стандартом для них [2].

R поддерживает широкий спектр статистических и численных методов и обладает хорошей расширяемостью с помощью пакетов. Пакеты представляют собой библиотеки для работы специфических функций или специальных областей применения. В базовую поставку R включен основной набор пакетов, а всего по состоянию на 2006 год доступно более 800 пакетов.

Ещё одной особенностью R являются графические возможности, заключающиеся в возможности создания качественной графики, которая может включать математические символы [2].

Установочные файлы последних версий для систем Windows и Unix можно найти на главной странице официального сайта среды разработки <http://cran.r-project.org/>[3]. Руководство по установке для операционных систем Windows и Unix находится в Приложении А. Пример тестовой программы для проверки работоспособности установленной системы находится в приводится в Приложении Б. Наиболее полным справочным ресурсом является сайт <http://www.statmethods.net/index.html>[4]. Краткие сведения по базовым функциям содержатся в ресурсе <http://cran.r-project.org/doc/contrib/Short-refcard.pdf>[5] и Приложении В.

Задание:

Установить и настроить IDE (Integrated Development Environment), интегрированную среду разработки для языка программирования R.

Результатом выполнения работы являются скриншоты последовательных шагов установки и проверка работоспособности IDE (например, программа, вычисляющая простейшее арифметическое выражение «2+2» или повтор примера в Приложении Б).

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 2. ОСНОВЫ РАБОТЫ СО СРЕДОЙ R

Цели работы: овладеть навыками программирования на R; изучить основные команды R; научиться устанавливать дополнительные пакеты для R, облегчающие разработку и позволяющие более наглядно представить результаты работы программ; разработать и проверить работоспособность тестовой программы.

Теоретические сведения.

R применяется везде, где нужна работа с данными. Это не только статистика в узком смысле слова, но и «первичный» анализ (графики, таблицы сопряжённости), и продвинутое математическое моделирование. R без особых проблем может использоваться и там, где сейчас принято использовать коммерческие программы анализа уровня MatLab/Octave. С другой стороны вполне естественно, что основная вычислительная мощь R лучше всего его проявляется при статистическом анализе: от вычисления средних величин до вейвлет-преобразований временных рядов. Особое место R занимает в Data Mining благодаря многим реализованным алгоритмам кластеризации и классификации, а так же бесчисленным способам визуализации полученных результатов [6].

R можно использовать в том числе и как обычный калькулятор. Например:

```
> 3+2
[1] 5
> 16+3/5-11*8^2
[1] -687.4
> (((16+3)/5)-11)*8)^2
[1] 3317.76
```

Для знакомых с интерактивными языками программирования типа Python здесь нет ничего особенно нового. Единственная деталь — это единичка в квадратных скобках. Она означает номер элемента вектора. Отсюда сразу два вывода:

1. R результат любой операции с числами трактует как вектор единичной длины. Скаляров в R просто нет;
2. Элементы векторов нумеруются с единицы, а не с нуля, как во многих языках программирования.

Просто открыть сессию R и вводить в окно программы команды, одну за другой — это лишь один из возможных способов работы. Гораздо более продуктивный метод, который является заодно и серьёзнейшим преимуществом R — это создание скриптов – программ, которые потом загружаются в R и интерпретируются им. С самого начала работы следует создавать скрипты, даже для несложных задач — это в будущем значительно экономит время.

Создать скрипт достаточно просто. Допустим, после открытия сессии была введена необходимая для получения искомого результата

последовательность команд. Чтобы сделать работу воспроизводимой, необходимо сохранить историю команд командой `savehistory`:

```
> savehistory(file="myscript.r")
```

После этого в текущей папке появляется файл `myscript.r`, который можно отредактировать любимом текстовом редакторе, а потом загрузить в R командой `source`:

```
> source("myscript.r", echo=TRUE)
```

Опция `echo` добавлена для того чтобы можно было видеть сами команды, а не только результат их выполнения.

Самое важное преимущество R — наличие многочисленных пакетов анализа и обработки данных, визуализации. При установке R на компьютер, несколько пакетов уже в наличии: так называемые базовые пакеты, без которых система просто не работает (например, пакет, который так и называется `base`, или пакет `grDevices`, который управляет выводом графиков), и ещё «рекомендованные» пакеты (пакет для специализированного кластерного анализа `cluster`, пакет для анализа нелинейных моделей `nlme` и пр.).

Кроме того, можно поставить любой из почти полутора тысяч доступных на CRAN пакетов. При доступном Интернете, это можно сделать прямо из R командой `install.packages()`. Как только пакет установлен, то он сразу готов к работе. Нужно только инициализировать его перед употреблением. Для этого служит команда `library()`. Более подробное руководство по установке содержится в приложении В.

Как и во многих языках программирования, в R существуют циклы и условные операторы.

Оператор `if`

Синтаксис оператора `if` очень прост:

```
if (условие){
```

```
  выполняется если условие верно
```

```
} else {
```

```
  выполняется если условие не верно
```

```
}
```

Оператор `ifelse()`

Команда `ifelse()` воспринимает первый аргумент как условие, второй аргумент возвращается если условие верно, а третий аргумент - если нет. Таким образом, условие может быть вектором. Например, мы генерируем последовательность от 1 до 10 и хотим вывести значения меньше чем 5 и больше чем 8.

```
> x <- 1:10
```

```
> ifelse(x<5 | x>8, x, 0)
```

```
[1] 1 2 3 4 0 0 0 0 9 10
```

R позволяет реализовать циклы тремя путями:

- циклы `for`;
- циклы с предусловием (`while`);
- «бесконечные» циклы (`repeat`);

Циклы for

Для использования оператора for требуется указать индекс (в примере – k) и вектор (в примере – 1 : 5), а также указать выполняемую конструкцию и указать повторяемые операторы в фигурных скобках. Механизм работы следующий:

1. интерпретатор последовательно выбирает значения из вектора и присваивает их переменной k;
2. с каждым значением k выполняется список функций в фигурных скобках.

Пример программы:

```
> for (k in 1:5){ print(k) }
```

```
[1] 1
```

```
[1] 2
```

```
[1] 3
```

```
[1] 4
```

```
[1] 5
```

Циклы с предусловием (while)

Синтаксис цикла while вполне стандартен:

1. Ключевое слово while.
2. Условие выполнимости в скобках.
3. Список функций для итерированного выполнения в фигурных скобках.

```
> g <- 0
```

```
> while (g < 1){
```

```
+   g <- rnorm(1)
```

```
+   cat(g, "\n")
```

```
+ }
```

```
-0.08111594
```

```
0.1732847
```

```
-0.2428368
```

```
0.3359238
```

```
-0.2080000
```

```
0.05458533
```

```
0.2627001
```

```
1.009195
```

«Бесконечные циклы» (repeat)

Нужно быть осторожным при использовании этого типа циклов, так как неправильное использование прерываний цикла (оператора break) может привести к бесконечному выполнению (зависанию) цикла. Синтаксис следующий:

1. Ключевое слово repeat.

2. Список функций для итерированного выполнения в фигурных скобках.

3. Наличие условия прерывания выполнения внутри цикла.

```
> repeat {
```

```
+   g <- rnorm(1)
```

```
+   if (g > 1.0) break
```



```
+   cat(g, "\n")
+   }
-1.214395
0.6393124
0.05505484
-1.217408
```

Использование методов кластеризации можно обобщить следующим образом:

1. Инициализация выборки (вектор или массив векторов).
2. Вызов метода применительно к выборке.
3. Визуализация результатов.[3]

Пример программы с использованием метода кластеризации h-кластеров:

```
> x<- matrix(c(86, 92, 94, 82, 92, 82, 82, 90, 89, 94, 95, 92, 92, 83, 88, 86,
95, 84, 90, 95, 9, 5, 10, 8, 5, 8, 5, 4, 2, 2, 7, 9, 9, 9, 10, 5, 9, 7, 7, 4), ncol = 2)
>hc <- hclust(x) #кластеризация h-кластерами
>plot(hc)
```

Задание:

- 1) Установить вспомогательный пакет для решения задач кластеризации (см. Приложение В).
- 2) Выполнить следующие действия:
 1. Создать массив целых чисел.
 2. Вычислить их сумму
 3. Считать несколько целых чисел из текстового файла.
 4. Конкатенировать два массива.
 5. С помощью оператора if определить чётная или нечётная длина полученного массива.
 6. Кластеризовать массив с помощью метода h-кластеров.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 3. МЕТОДЫ ВИЗУАЛИЗАЦИИ В СРЕДЕ R

Цель работы: изучить основные методы визуализации; программно отобразить некоторые из них, согласно варианту.

Теоретические сведения.

Визуализация – это инструментарий, который позволяет увидеть конечный результат вычислений, организовать управление вычислительным процессом и даже вернуться назад к исходным данным, чтобы определить наиболее рациональное направление дальнейшего движения [7].

В результате использования визуализации создается графический образ данных. Применение визуализации помогает в процессе анализа данных увидеть аномалии, структуры, тренды.

В данной лабораторной работе будут рассмотрены 6 основных типов представления результатов кластеризации: дендрограмма, график компонентов, гистограммы, контурный график, график центров кластеров, box-график.

Дендрограмма

Текст программы:

```
mydata <- matrix(c(86, 92, 94, 82, 92, 82, 82, 90, 89, 94, 95, 92, 92, 83, 88, 86, 95,  
84, 90, 95, 9, 5, 10, 8, 5, 8, 5, 4, 2, 2, 7, 9, 9, 9, 10, 5, 9, 7, 7, 4), ncol = 2)  
h <- hclust(dist(mydata)) #кластеризация h-кластерами plot(h)
```

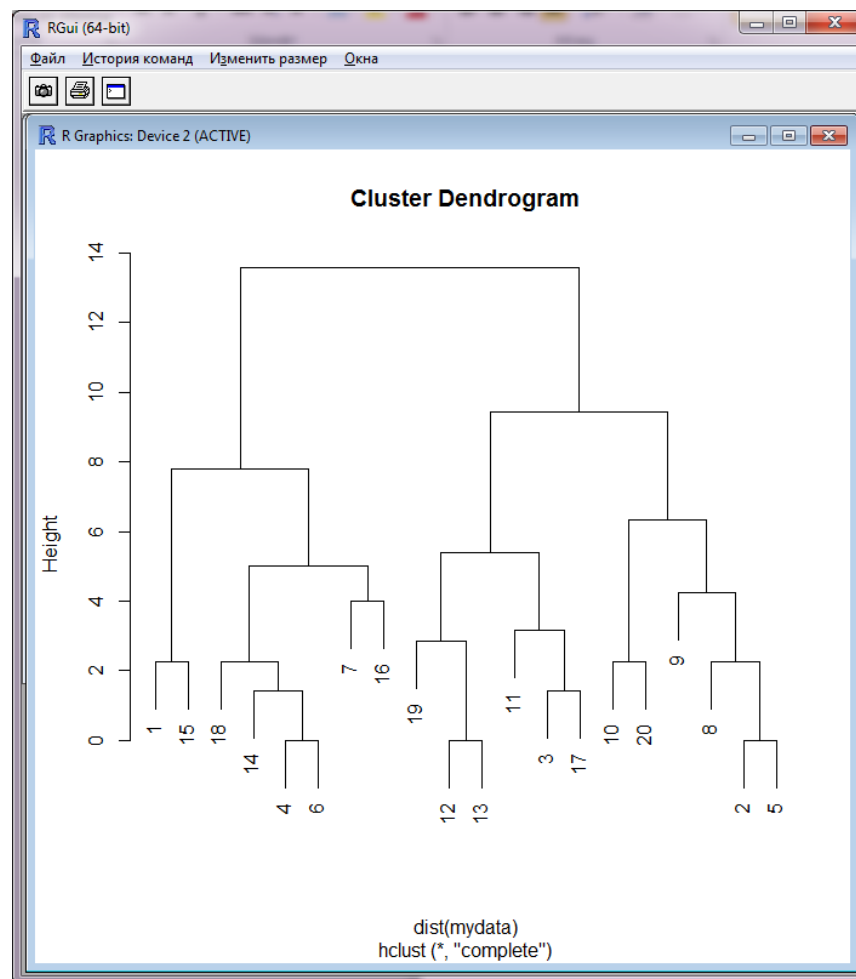


Рис.1. Дендрограмма.

График компонентов

Текст программы:

```
mydata <- matrix(c(86, 92, 94, 82, 92, 82, 82, 90, 89, 94, 95, 92, 92, 83, 88, 86, 95,
84, 90, 95, 9, 5, 10, 8, 5, 8, 5, 4, 2, 2, 7, 9, 9, 9, 10, 5, 9, 7, 7, 4), ncol = 2)
```

```
fit <- kmeans(mydata, 5)
```

```
library(cluster)
```

```
clusplot(mydata, fit$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```

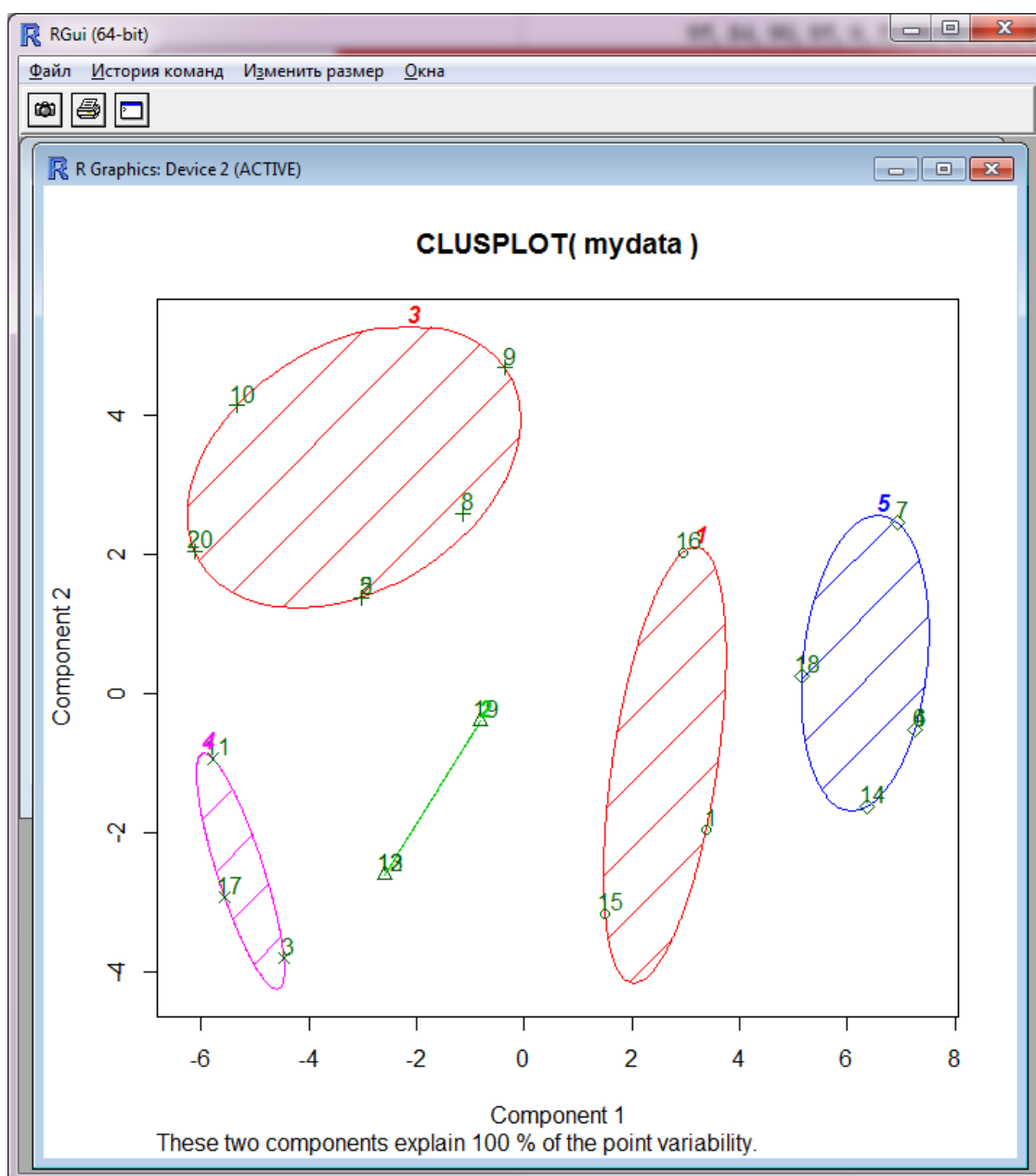


Рис. 2. График компонент.

Гистограммы

Текст программы:

```
mydata <- matrix(c(86, 92, 94, 82, 92, 82, 82, 90, 89, 94, 95, 92, 92, 83, 88, 86, 95,
84, 90, 95, 9, 5, 10, 8, 5, 8, 5, 4, 2, 2, 7, 9, 9, 9, 10, 5, 9, 7, 7, 4), ncol = 2)
hist(mydata)
```

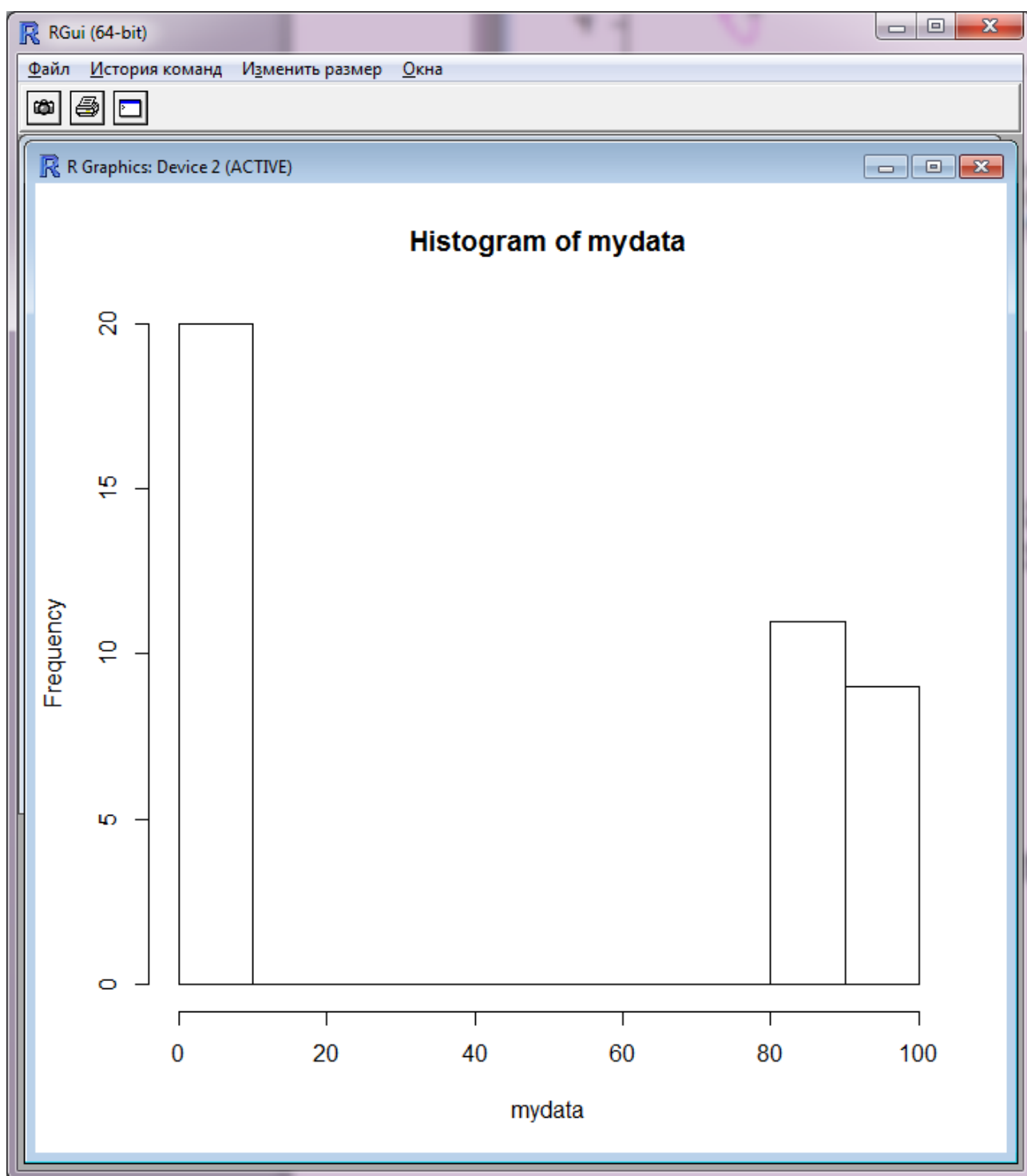


Рис. 3. Гистограммы.

Контурный график

Текст программы:

```
library(mclust)
```

```
mydata <- read.table('C:/Users/on/Desktop/klaster/cs.dat', sep=',', header=TRUE)
```

```
#чтение из файла данных через запятую
```

```
fit <- Mclust(mydata)
```

```
plot(fit, mydata)
```

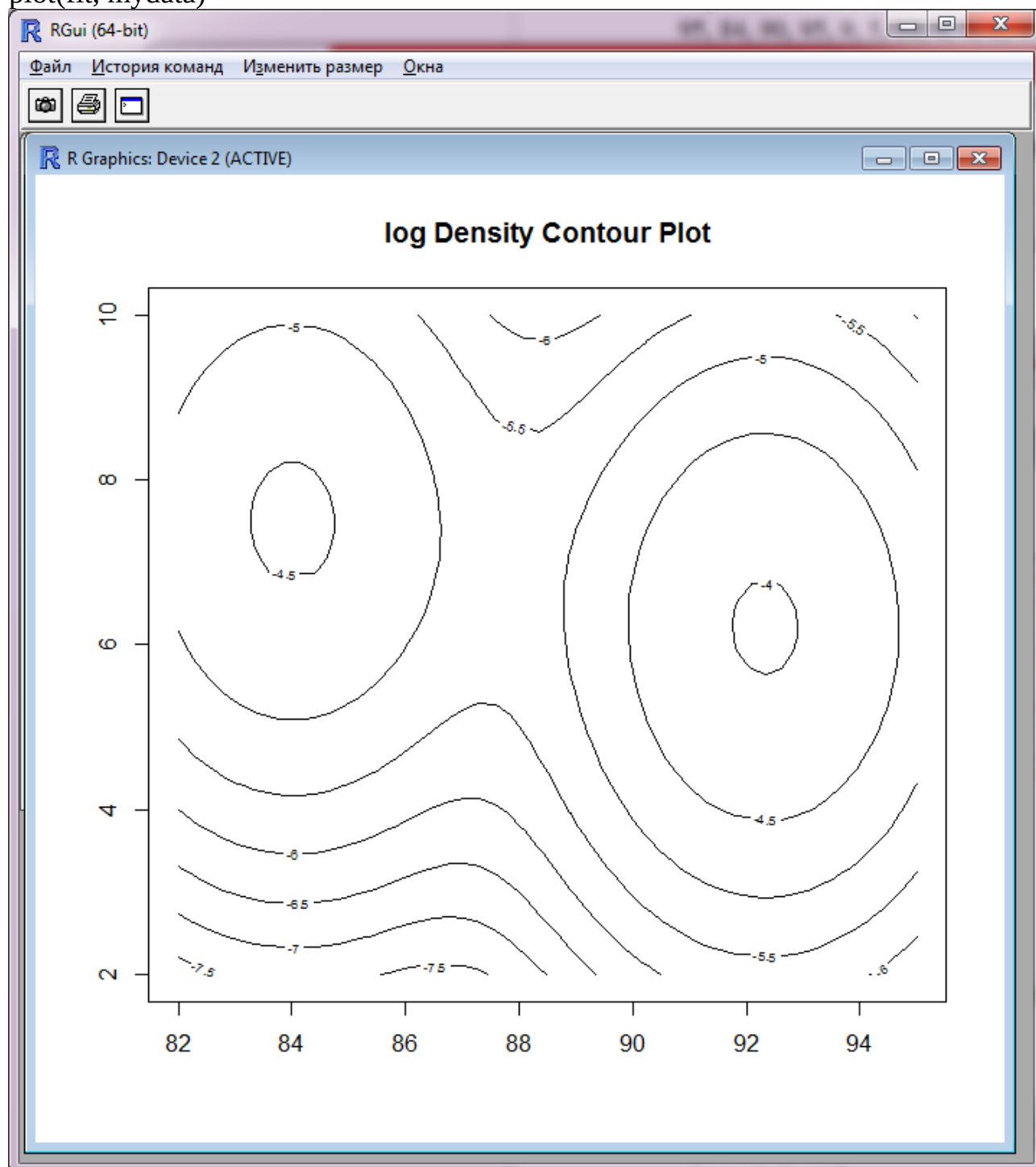


Рис. 4. Контурный график.

График центров кластеров

Текст программы:

```
cells <- c(1, 1, 2, 1, 4, 3, 5, 4)
```

```
rnames <- c("A", "B", "C", "D")
```

```
cnames <- c("X", "Y")
```

```
x <- matrix(cells, nrow=4, ncol=2, byrow=TRUE, dimnames=list(rnames, cnames))
```

```
km <- kmeans(x, 2, 15)
```

```
plot(x, col = km$cluster)
```

```
points(km$centers, col = 1:2, pch = 8)
```

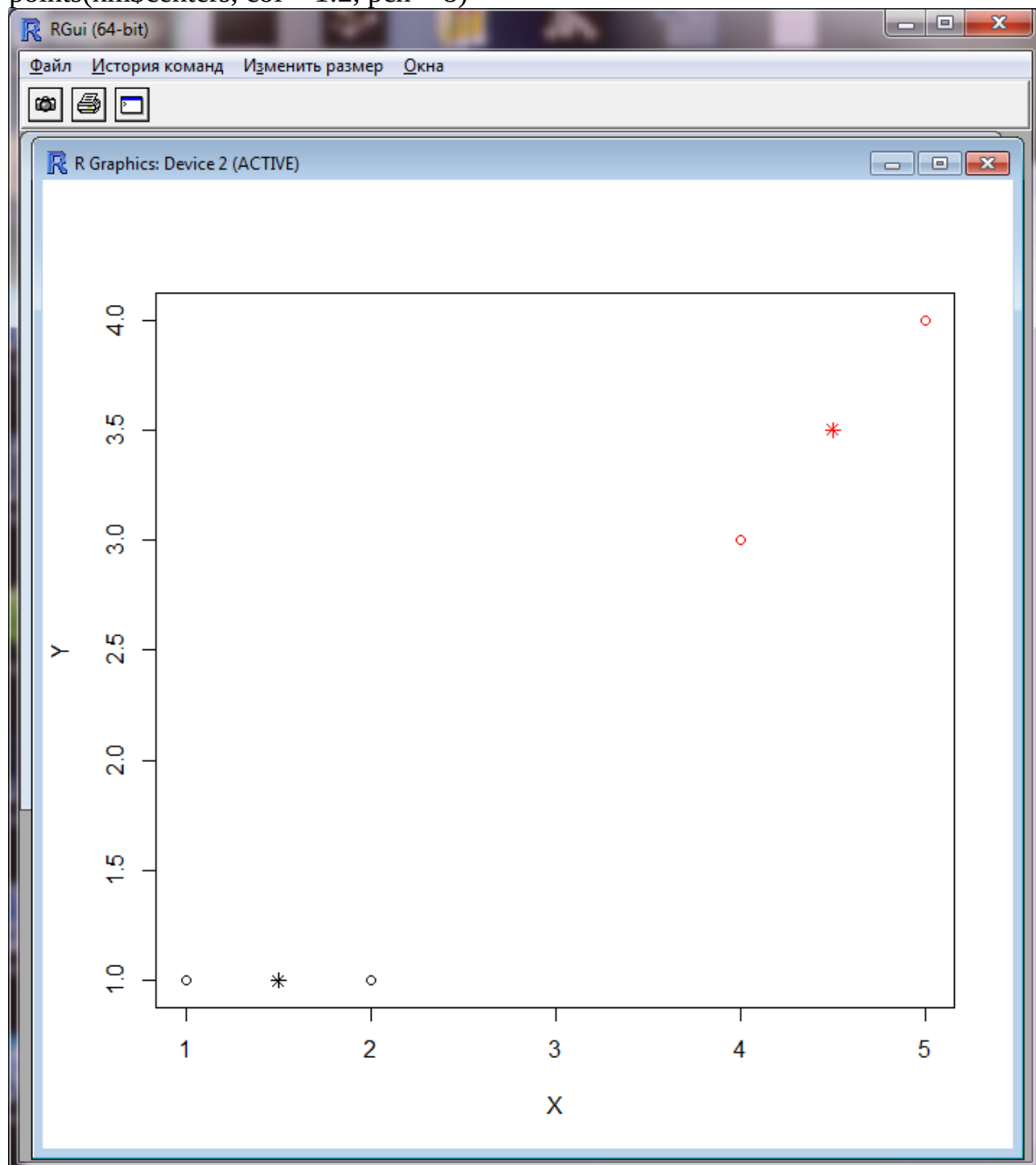


Рис. 4. График центров кластеров.

Вох-график

Текст программы:

```
x <- rlbmat()
pr <- proximus(x, max.radius=8, debug=TRUE)
op <- par(mfrow=c(1,2), pty="s")
lplot(x, main="Data")
box()
lplot(fitted(pr)$x, main="Approximation")
box()
par(op)
```

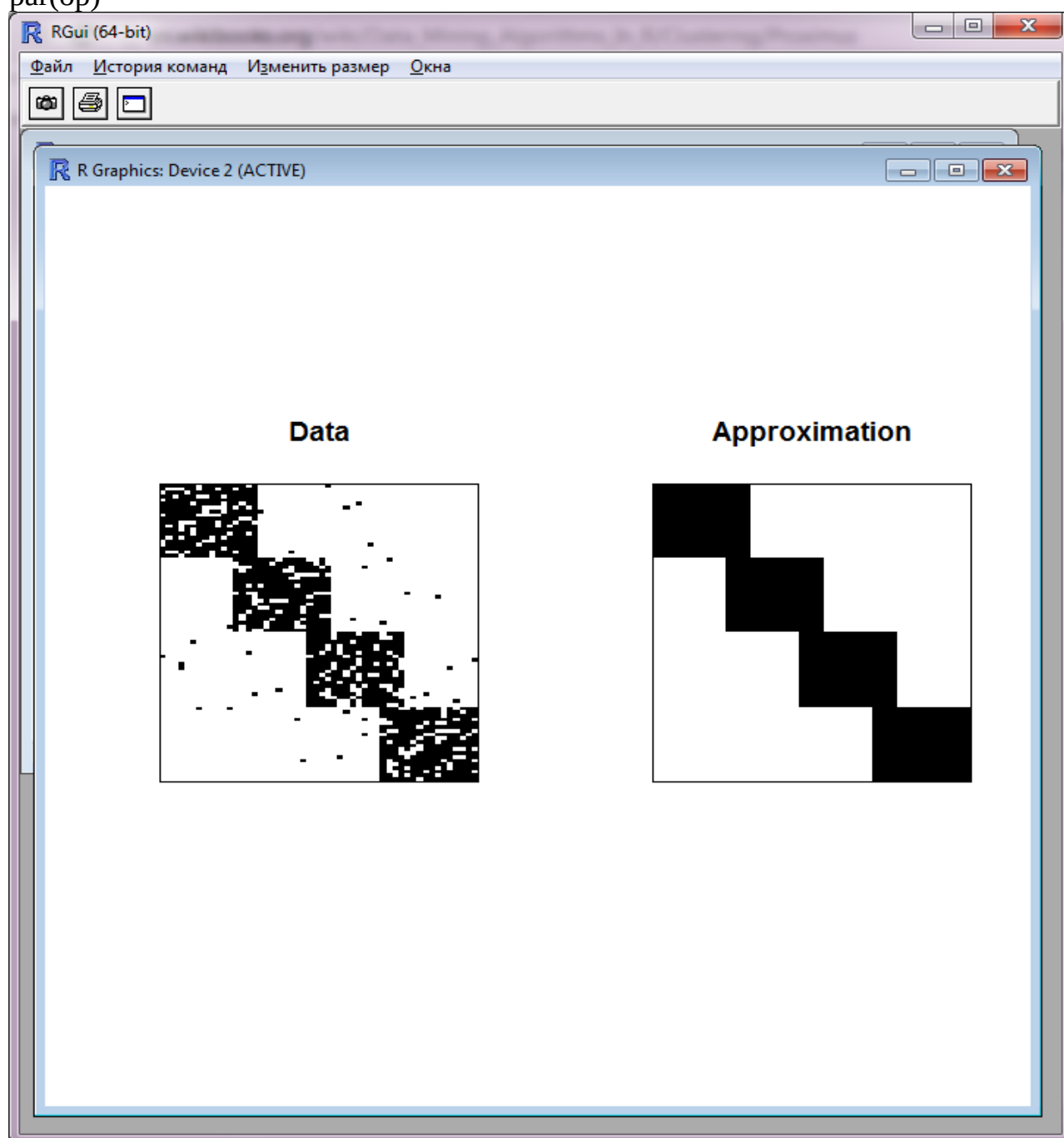


Рис. 5. Вох-график.

Задание:

Установить вспомогательный пакет для решения задач кластеризации. Кластеризовать контрольную выборку методом МВС (Model Based Clustering) или k-средних. Результаты кластеризации визуализировать согласно варианту.

Варианты:

1. Дендрограмма и контурный график.
2. График компонентов и Дендрограмма.
3. Гистограммы и Вох-график.
4. Контурный график и график центров кластеров.
5. График центров кластеров и гистограммы.
6. Вох-график и график компонентов.
7. Дендрограмма и Вох-график.
8. График компонентов и контурный график.
9. Гистограммы и график центров кластеров.
10. Контурный график и график компонентов.
11. График центров кластеров и Дендрограмма.
12. Вох-график и гистограммы.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 4. КЛАССИФИКАЦИЯ.

Цели работы: изучить понятие «классификация»; изучить наиболее популярные методы классификации; научиться использовать методы классификации среды R.

Теоретические сведения.

Классификация (Classification) – наиболее простая и распространенная задача Data Mining. В результате решения задачи классификации обнаруживаются признаки, которые характеризуют группы объектов исследуемого набора данных – классы. По этим признакам новый объект можно отнести к тому или иному классу.[8]

В описательных задачах самое главное — это дать наглядное описание имеющихся скрытых закономерностей, в то время как в предсказательных задачах на первом плане стоит вопрос о предсказании для тех случаев, для которых данных ещё нет.

К описательным задачам относятся:

- поиск ассоциативных правил или паттернов (образцов);
- группировка объектов, кластерный анализ;
- построение регрессионной модели.

К предсказательным задачам относятся:

- классификация объектов (для заранее заданных классов);
- регрессионный анализ, анализ временных рядов

Классификация требует соблюдения следующих правил:

- в каждом акте деления необходимо применять только одно основание;
- деление должно быть соразмерным, т.е. общий объем видовых понятий должен равняться объему делимого родового понятия;
- члены деления должны взаимно исключать друг друга, их объемы не должны перекрещиваться;
- деление должно быть последовательным.

Различают:

- вспомогательную (искусственную) классификацию, которая производится по внешнему признаку и служит для придания множеству предметов (процессов, явлений) нужного порядка;
- естественную классификацию, которая производится по существенным признакам, характеризующим внутреннюю общность предметов и явлений. Она является результатом и важным средством научного исследования, т.к. предполагает и закрепляет результаты изучения закономерностей классифицируемых объектов.[9]

В зависимости от выбранных признаков, их сочетания и процедуры деления понятий классификация может быть:

- простой – деление родового понятия только по признаку и только один раз до раскрытия всех видов. Примером такой классификации является дихотомия, при которой членами деления бывают только два понятия, каждое из которых является противоречащим другому (т.е. соблюдается принцип: «А и не А»);

- сложной – применяется для деления одного понятия по разным основаниям и синтеза таких простых делений в единое целое. Примером такой классификации является периодическая система химических элементов.

Под классификацией будем понимать отнесение объектов (наблюдений, событий) к одному из заранее известных классов.

Задачей классификации часто называют предсказание категориальной зависимой переменной (т.е. зависимой переменной, являющейся категорией) на основе выборки непрерывных и/или категориальных переменных.

Например, можно предсказать, кто из клиентов фирмы является потенциальным покупателем определенного товара, а кто – нет, кто воспользуется услугой фирмы, а кто – нет, и т.д. Этот тип задач относится к задачам бинарной классификации, в них зависимая переменная может принимать только два значения (например, да или нет, 0 или 1).

Другой вариант классификации возникает, если зависимая переменная может принимать значения из некоторого множества предопределенных классов. Например, когда необходимо предсказать, какую марку автомобиля захочет купить клиент. В этих случаях рассматривается множество классов для зависимой переменной [10].

Классификация может быть одномерной (по одному признаку) и многомерной (по двум и более признакам).

Для классификации в DataMining используется множество различных моделей: нейронные сети, деревья решений, машины опорных векторов, алгоритмы покрытия и др., при построении которых применяется обучение с учителем, когда выходная переменная (метка класса) задана для каждого наблюдения [11].

Задание:

Установить вспомогательные пакеты для решения задач классификации ("e1071", "RWeka", "mvoutlier", "rpart", "class", "ada", "caret"). Классифицировать контрольную выборку заданным методом согласно варианту. Результаты визуализировать способом, стандартным для используемого пакета.

Варианты:

1. SVM и JRip
2. penalizedSVM и kNN
3. kNN и Outliers
4. SVM и Outliers
5. Outliers и Adaboost
6. Decision Trees и JRip
7. SVM и Naive Bayes
8. Naive Bayes и
9. Adaboost и Outliers
10. JRip и kNN
11. Outliers и JRip
12. SVM и Adaboost

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 5. КЛАСТЕРИЗАЦИЯ

Цели работы: изучить наиболее популярные методы и алгоритмы кластеризации; запрограммировать алгоритмы кластеризации и визуализировать результаты.

Теоретические сведения.

Кластерный анализ – это способ группировки многомерных объектов, основанный на представлении результатов отдельных наблюдений точками подходящего геометрического пространства с последующим выделением групп как «сгустков» этих точек (кластеров, таксонов). Данный метод исследования получил развитие в последние годы в связи с возможностью компьютерной обработки больших баз данных [11]. Кластерный анализ предполагает выделение компактных, удаленных друг от друга групп объектов, отыскивает «естественное» разбиение совокупности на области скопления объектов. Он используется, когда исходные данные представлены в виде матриц близости или расстояний между объектами либо в виде точек в многомерном пространстве [12]. Наиболее распространены данные второго вида, для которых кластерный анализ ориентирован на выделение некоторых геометрически удаленных групп, внутри которых объекты близки. Выбор расстояния между объектами является узловым моментом исследования, от него во многом зависит окончательный вариант разбиения объектов на классы при данном алгоритме разбиения.

Существует большое количество алгоритмов кластерного анализа, их можно разделить по способу построения кластеров на 2 типа: эталонные и неэталонные. В процедурах эталонного типа на множестве объектов задается несколько исходных зон, с которых начинает работу алгоритм. Эталоны могут представлять собой первоначальное разбиение на классы, центр тяжести класса и др. После задания эталонов алгоритм производит классификацию, иногда меняя определенным способом эталоны. К алгоритмам кластеризации, работающим по иному принципу, относятся иерархические алгоритмы кластерного анализа, процедура разрезания и др. [13].

Решением задачи кластерного анализа является разбиение, удовлетворяющее некоторому условию оптимальности. Этот критерий может представлять собой некоторый функционал, выражающий уровни желательности различных разбиений и группировок. Этот функционал часто называют целевой функцией. Задачей кластерного анализа является задача оптимизации, т.е. нахождение минимума целевой функции при некотором заданном наборе ограничений. Примером целевой функции может служить, в частности, сумма квадратов внутригрупповых отклонений по всем кластерам.

Рассмотрим основные способы определения расстояний между объектами. Попадание в один или разные кластеры объектов определяется понятием расстояния между X_i и X_j из E_p , где E_p - p -мерное евклидово

пространство. Неотрицательная функция $d(X_i, X_j)$ называется функцией расстояния (метрикой), если:

- а) $d(X_i, X_j) \geq 0$, для всех X_i и X_j из E_p ;
- б) $d(X_i, X_j) = 0$, тогда и только тогда, когда $X_i = X_j$
- в) $d(X_i, X_j) = d(X_j, X_i)$
- г) $d(X_i, X_j) \leq d(X_i, X_k) + d(X_k, X_j)$, где X_j ; X_i и X_k - любые три вектора из E_p .

Значение $d(X_i, X_j)$ для X_i и X_j называется расстоянием между X_i и X_j и эквивалентно расстоянию между G_i и G_j соответственно выбранным характеристикам $(F_1, F_2, F_3, \dots, F_p)$.

Наиболее часто используемые виды метрик для количественных шкал [10] приведены в Таблице 1.

Таблица 1 - Основные виды метрик для количественных шкал.

Название	Расчетная формула
Евклидово расстояние	$d_2(x_i, x_j) = \left[\sum_{k=1}^p (x_{ki} - x_{kj})^2 \right]^{\frac{1}{2}}$
Линейное расстояние («хэмминговое, манхеттенское»)	$d_1(x_i, x_j) = \left[\sum_{k=1}^p x_{ki} - x_{kj} \right]$
Степенное расстояние	$d_p(x_i, x_j) = \left[\sum_{k=1}^p x_{ki} - x_{kj} ^p \right]^{\frac{1}{p}}$

Пусть n измерений $X_1, X_2, \dots, X_n$ представлены в виде матрицы данных (обучающей выборки) размером $p \times n$:

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{p1} & x_{p2} & \dots & x_{pn} \end{pmatrix} = (x_1, x_2, \dots, x_n)$$

Тогда расстояние между парами векторов $d(X_i, X_j)$ могут быть представлены в виде симметричной матрицы расстояний:

$$D = \begin{pmatrix} 0 & d_{12} & \dots & d_{1n} \\ d_{21} & 0 & \dots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \dots & 0 \end{pmatrix}$$

Понятием, противоположным расстоянию, является понятие сходства между объектами G_i и G_j . Неотрицательная вещественная функция $S(X_i, X_j) = S_{ij}$ называется мерой сходства, если:

- 1) $0 \leq S(X_i, X_j) < 1$ для $X_i \neq X_j$,
- 2) $S(X_i, X_j) = 1$,
- 3) $S(X_i, X_j) = S(X_j, X_i)$.

Пары значений мер сходства можно объединить в матрицу сходства:

$$S = \begin{pmatrix} 1 & S_{12} & \dots & S_{1n} \\ S_{21} & 1 & \dots & S_{2n} \\ S_{n1} & S_{n2} & \dots & 1 \end{pmatrix}$$

Величину S_{ij} называют коэффициентом сходства. Связь между коэффициентом сходства и расстоянием определяется формулой

$$S_{ij} = \frac{1}{1 + d_{ij}}$$

Известны следующие математические характеристики кластера:

Центр кластера – это среднее геометрическое место точек в пространстве переменных. Радиус кластера – максимальное расстояние точек от центра кластера. Размер кластера может быть определен либо по радиусу кластера, либо по среднеквадратичному отклонению объектов для этого кластера.

Объект относится к кластеру, если расстояние от объекта до центра кластера меньше радиуса кластера. В настоящее время процедуры эталонного типа применимы для решения многих задач классификации, алгоритмы быстры и удобны в вычислительном отношении, их результаты наглядно представимы в диаграммах и графиках. Для проведения эталонной классификации необходимо выбрать метод первичного задания эталонных множеств и способ корректировки классов и стабилизации в целом, задать значения параметров алгоритма кластеризации. Иерархические алгоритмы кластерного анализа могут быть двух типов – агломеративные и дивизионные. В агломеративных процедурах начальным является разбиение, состоящее из n одноэлементных классов, а конечным – из одного класса, в дивизионных – наоборот. Принцип работы иерархических агломеративных (дивизионных) алгоритмов состоит в последовательном объединении (разделении) групп элементов, т.е. в создании иерархической структуры классов. Обычно такая классификация представляется в виде дендограммы – графика, отражающего последовательное объединение двух кластеров в один с указанием расстояний между ними.

Варианты заданий:

1. Кластеризовать выборку средствами языка R методом k-средних с указанием числа кластеров. Выбрать 3 различных числа ожидаемых кластеров. Результаты представить в виде графика компонентов и проанализировать.

2. Кластеризовать выборку средствами языка R методом k-средних с использованием Манхеттенской и Эвклидовой метрик. Результаты представить в виде графика компонентов и сравнить полученные результаты.
3. Кластеризовать выборку средствами языка R методом MBC (Model Based Clustering). Результаты представить в виде Classification и контурного графика, сравнить полученные результаты.
4. Выполнить кластеризацию выборку средствами языка R методом Варда с использованием эвклидовой метрики и методом h-кластеров. Результаты представить в виде дендрограмм и сравнить полученные визуализации.
5. Кластеризовать выборку согласно средствами языка R методом k-средних с использованием Степенной и Эвклидовой метрик. Результаты представить в виде графика компонентов и сравнить полученные результаты.
6. Выполнить кластеризацию выборки средствами языка R методом k-средних с использованием эвклидовой метрики и методом h-кластеров. Результаты представить в виде дендрограмм и сравнить полученные визуализации.
7. Кластеризовать выборку средствами языка R методом k-средних с указанием числа кластеров. Выбрать 3 различных числа ожидаемых кластеров. Результаты представить в виде дендрограммы и проанализировать.
8. Кластеризовать выборку средствами языка R методом Варда. Результаты представить в виде дендограммы и контурного графика, сравнить полученные результаты.
9. Кластеризовать выборку средствами языка R методом Bayesian Hierarchical Clustering и методом h-кластеров. Результаты представить в виде дендограмм, сравнить полученные результаты.
10. Кластеризовать выборку средствами языка R методом CLARA с использованием Степенной и Эвклидовой метрик. Результаты представить в виде дендрограмм и сравнить полученные результаты.
11. Кластеризовать выборку средствами языка R методом Biclust с указанием числа кластеров. Выбрать 3 различных числа ожидаемых кластеров. Результаты представить в виде дендрограммы и проанализировать.
12. Кластеризовать выборку средствами языка R методом Варда и c-средних. Результаты представить в виде дендрограмм, сравнить полученные результаты.

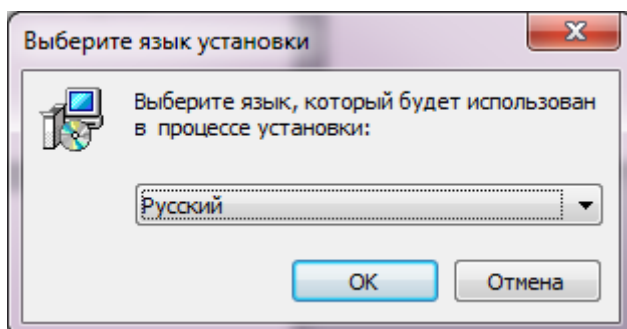
БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Управление знаниями. [Электронный ресурс]. – Электрон. Текстовые данные (6166 bytes). – М.: «Вильямс», 2001. - Режим доступа: http://msk.treko.ru/show_dict_390 Monday, 15 May 2001 14:17:13.
- 2 Виттих, В. А., Интеллектуальный анализ данных с помощью кластеризации. [Электронный ресурс]. – Электрон. Текстовые данные (14113 bytes). – М.: В. А., Виттих, И. В. Майоров, П. О. Скобелев, О. Л. Сурнин. Режим доступа: <http://www.kg.ru/support/library/clustering/>
- 3 the Comprehensive R Archive Network. [Электронный ресурс]. – Электрон. Текстовые данные (1141 bytes). – М.: Institute of Statistics and Mathematics, 2013. – Режим доступа: <http://cran.r-project.org/>
- 4 About Quick-R [Электронный ресурс]. – Электрон. Текстовые данные (8909 bytes). – М.: Robert I. Kabacoff, Ph.D. Режим доступа: <http://www.statmethods.net/index.html>
- 5 R Reference Card. [Электронный ресурс]. – Электрон. Текстовые данные (73028 bytes). – М.: Tom Short, EPRI PEAC. Режим доступа: <http://cran.r-project.org/doc/contrib/Short-refcard.pdf>
- 6 Jain A., Murty M., Flynn P. Data clustering: A review // ACM Computing Surveys. – 1999. - Vol. 31, no. 3. - Pp. 264–323.
- 7 Lance, G. N. A general theory of classification sorting strategies. Hierarchical [Электронный ресурс]. – Электрон. Текстовые данные (12 megabytes). – М.: G. N. Lance, W. T. Williams. Режим доступа: <http://comjnl.oxfordjournals.org/>
- 8 Баранов, В.В. Процессы принятия управленческих решений, мотивированных интересами / В.В. Баранов. – М: ФИЗМАТЛИТ, 2005. – 296 с.
- 9 Брукс, П. Метрики для управления ИТ-услугами: пер. с англ. / Питер Брукс. – М: Альпина Бизнес Букс, 2008. – 283 с.
- 10 Бон, Ян Ван. ИТ Сервис-менеджмент, введение // Я.В. Бон, Г. Кеммерлинг, Д. Пондаман; под ред. М.Ю. Потоцкого (русская версия). – М: IT Expert, 2003. – 215 с.
- 11 Чубукова, И. А. Data Mining: учебное пособие / И. А. Чубукова. — М.: Интернет-университет информационных технологий: БИНОМ: Лаборатория знаний, 2006. — 382 с. — ISBN 5-9556-0064-7.
- 12 Мандель, И. Д. Кластерный анализ. / И. Д. Мандель. — М.: Финансы и статистика, 1988 – 176 с.

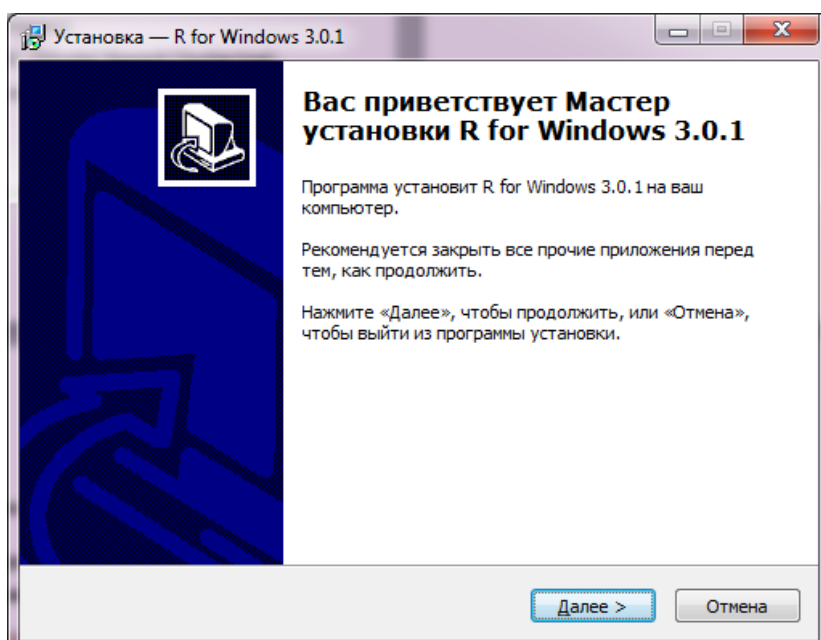
Приложение А

Установка среды разработки на R для Windows 7.

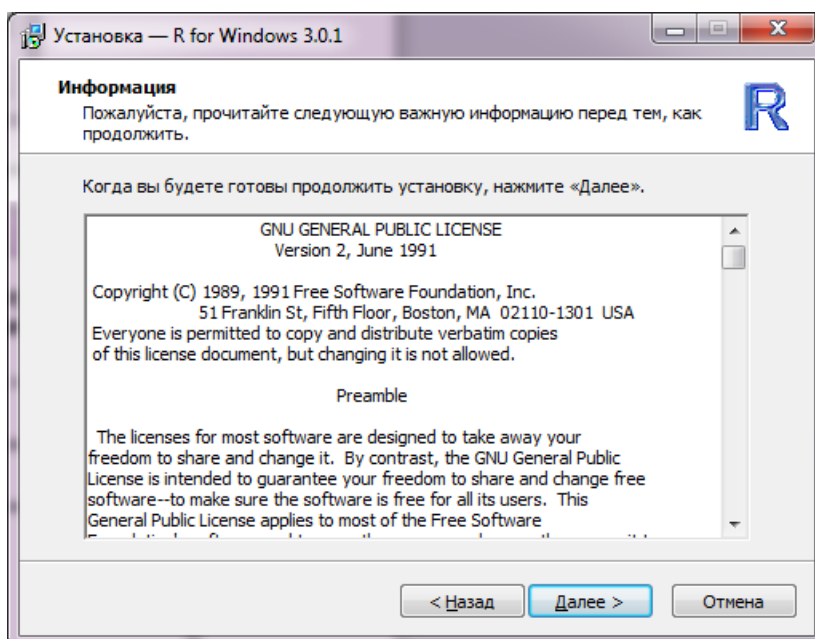
1. Скачать установочный файл последней версии (на текущий момент 3.0.1).
2. Запустить процесс установки. Выбрать русский язык.



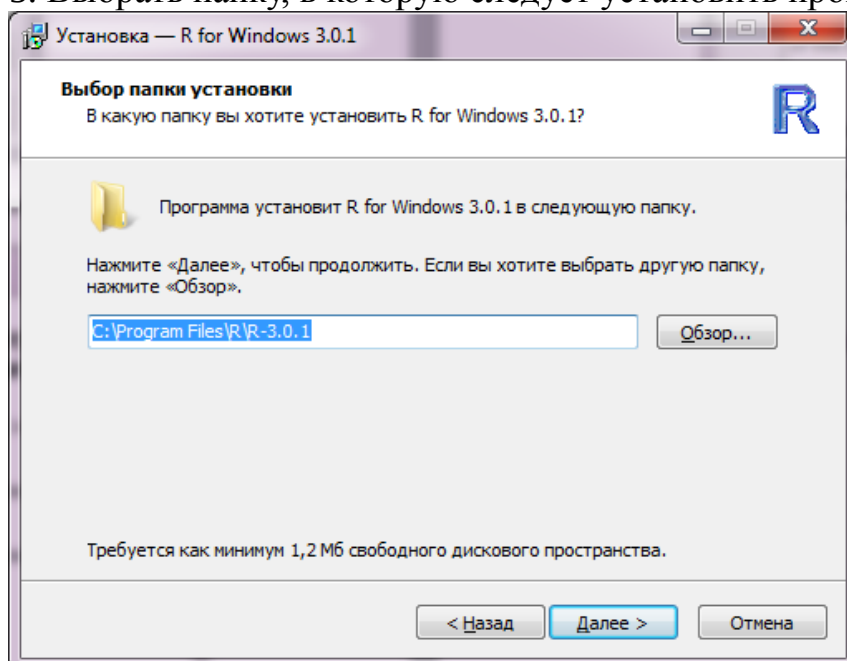
3. В следующем окне нажать кнопку Далее>.



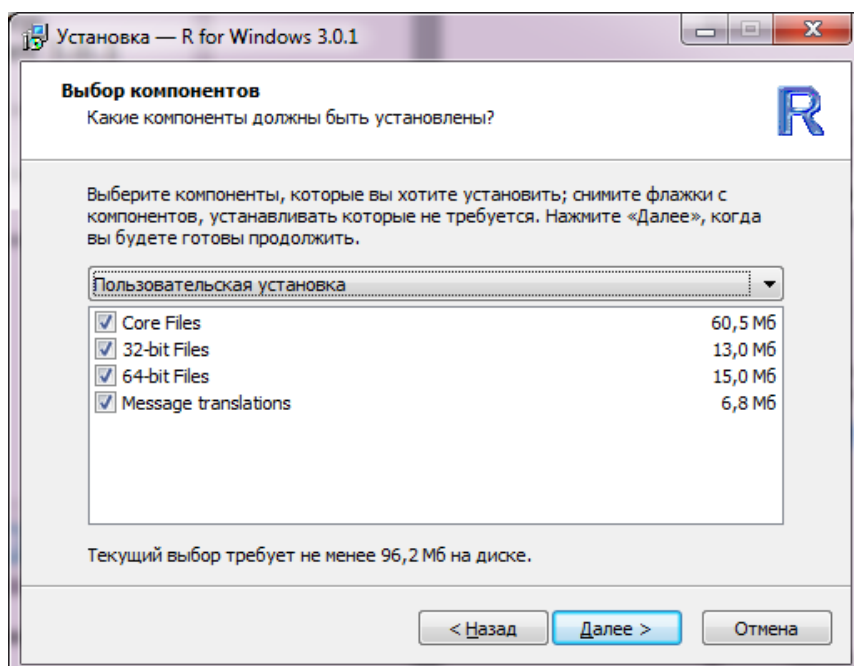
4. В следующем окне нажать кнопку Далее>.



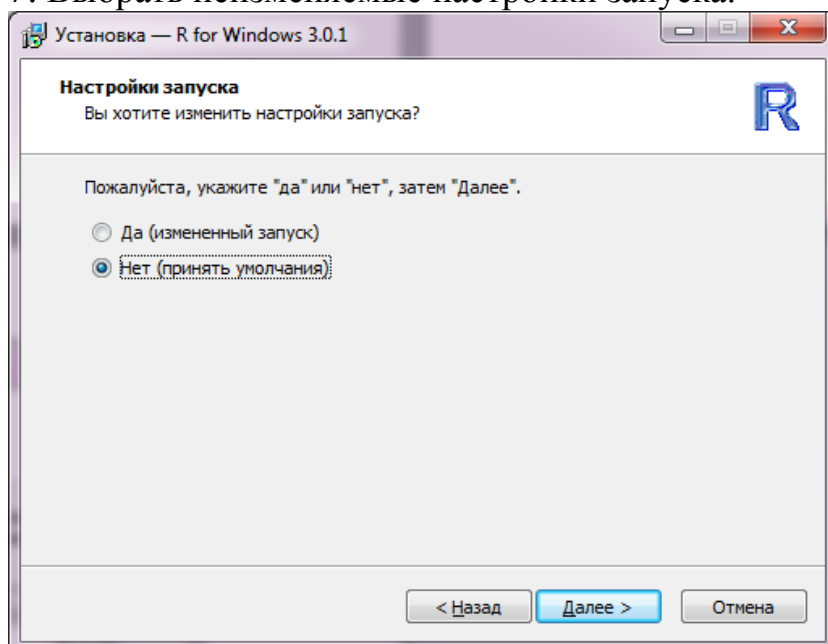
5. Выбрать папку, в которую следует установить программу.



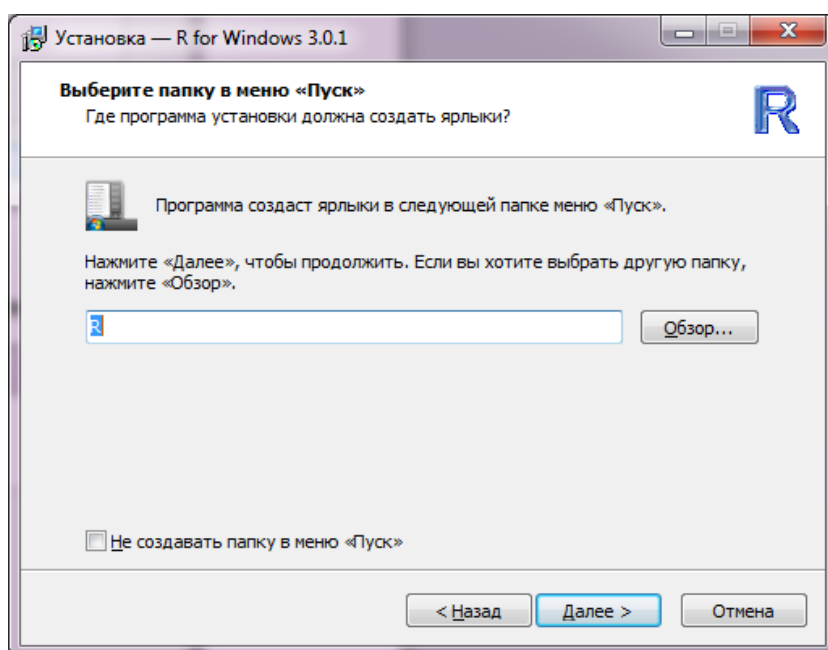
6. В следующем окне выбрать Пользовательскую установку и нажать кнопку Далее>.



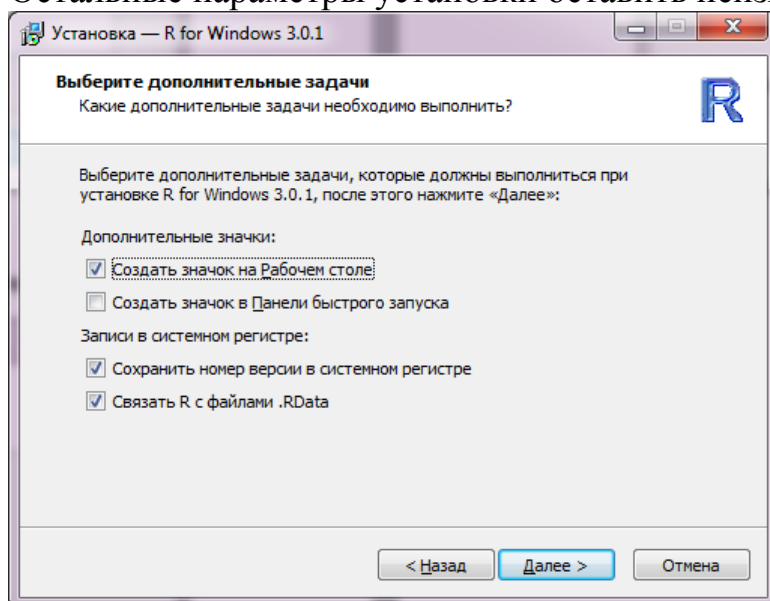
7. Выбрать неизменяемые настройки запуска.



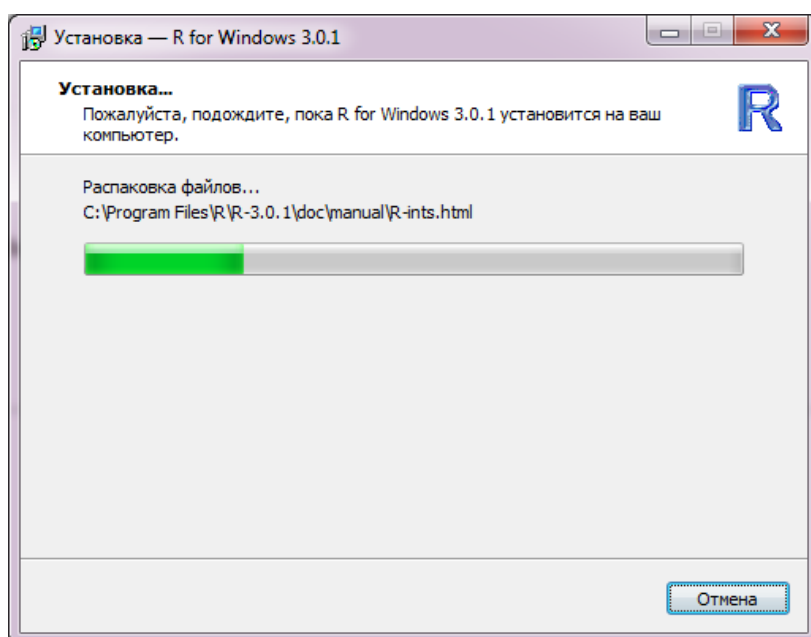
8. По желанию можно не создавать ярлык в меню Пуск.



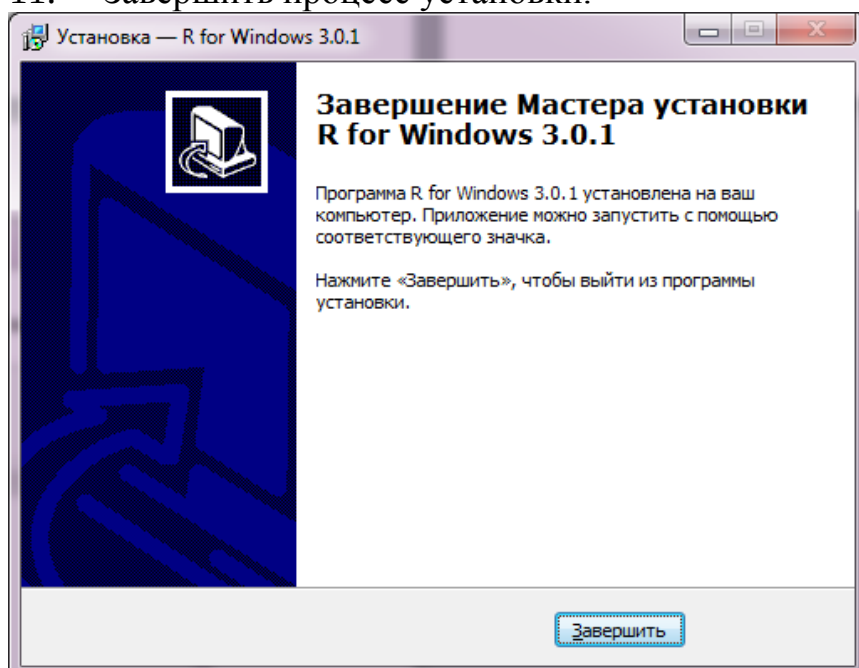
9. По желанию можно создать ярлык на Рабочем столе компьютера. Остальные параметры установки оставить неизменными.



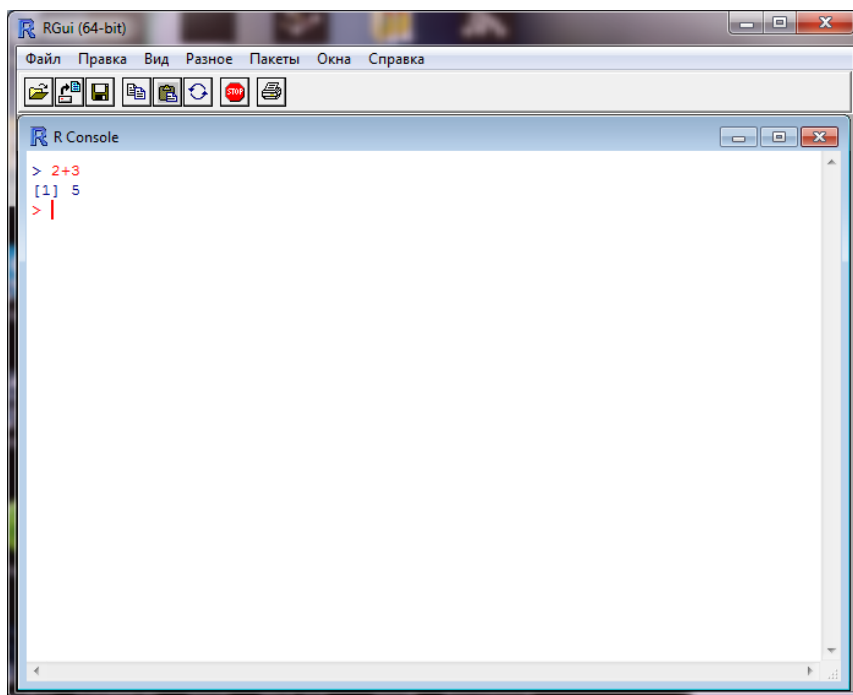
10. Нажать кнопку **Далее>** и запустить процесс установки.



11. Завершить процесс установки.



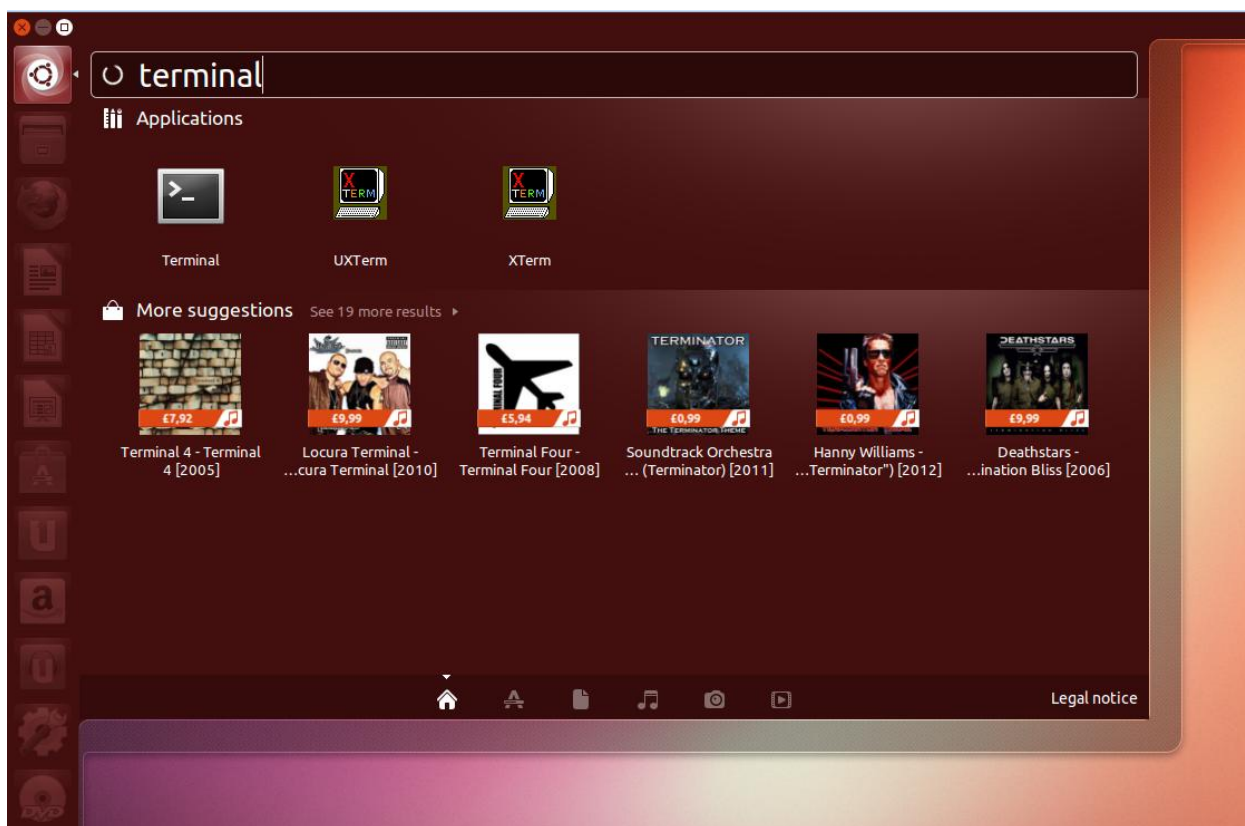
12. Запустить установленную программу и проверить её работоспособность.



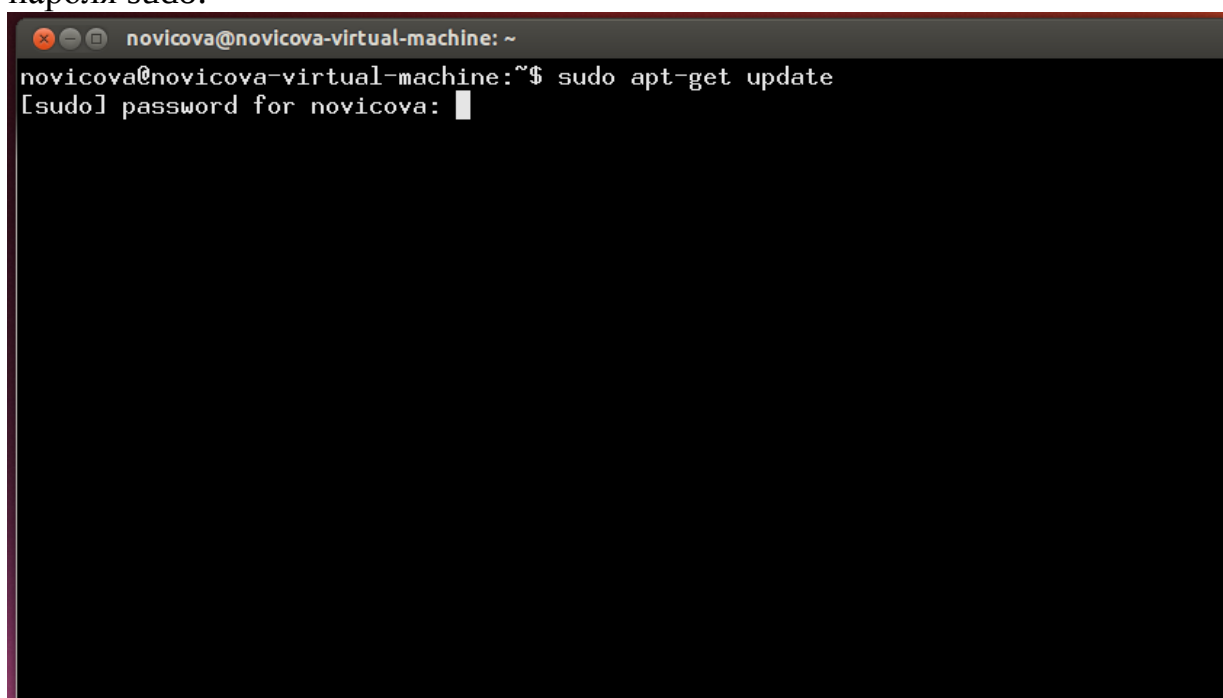
Приложение Б

Установка R в Ubuntu 13.04 Raring Ringtail.

1. Нажав на Dash необходимо запустить приложение «Terminal»:



2. Так как R доступен из репозитория Ubuntu, начиная с версии 10.04, необходимо обновить список доступных пакетов, подтвердив команду вводом пароля sudo:



3. После обновления списка доступных пакетов необходимо установить пакет `r-base` и все его зависимости:

```
novicova@novicova-virtual-machine: ~  
novicova@novicova-virtual-machine:~$ sudo apt-get install r-base -y
```

4. После того, как установка завершится, можно запустить интерактивную оболочку R и проверить её работоспособность:

```
novicova@novicova-virtual-machine: ~  
novicova@novicova-virtual-machine:~$ R  
  
R version 2.15.2 (2012-10-26) -- "Trick or Treat"  
Copyright (C) 2012 The R Foundation for Statistical Computing  
ISBN 3-900051-07-0  
Platform: i686-pc-linux-gnu (32-bit)  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.  
  
Natural language support but running in an English locale  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
> 2+2  
[1] 4  
>
```


Приложение В

Пример проверочной программы для лабораторной работы №1.

```
m <- c(3, 5, 5, 3, 4, 6, 3, 1, 6, 7, 8, 4, 3, 6, 8, 9, 5, 2, 3, 4, 5, 6, 7, 3, 2, 5, 6, 2)
```

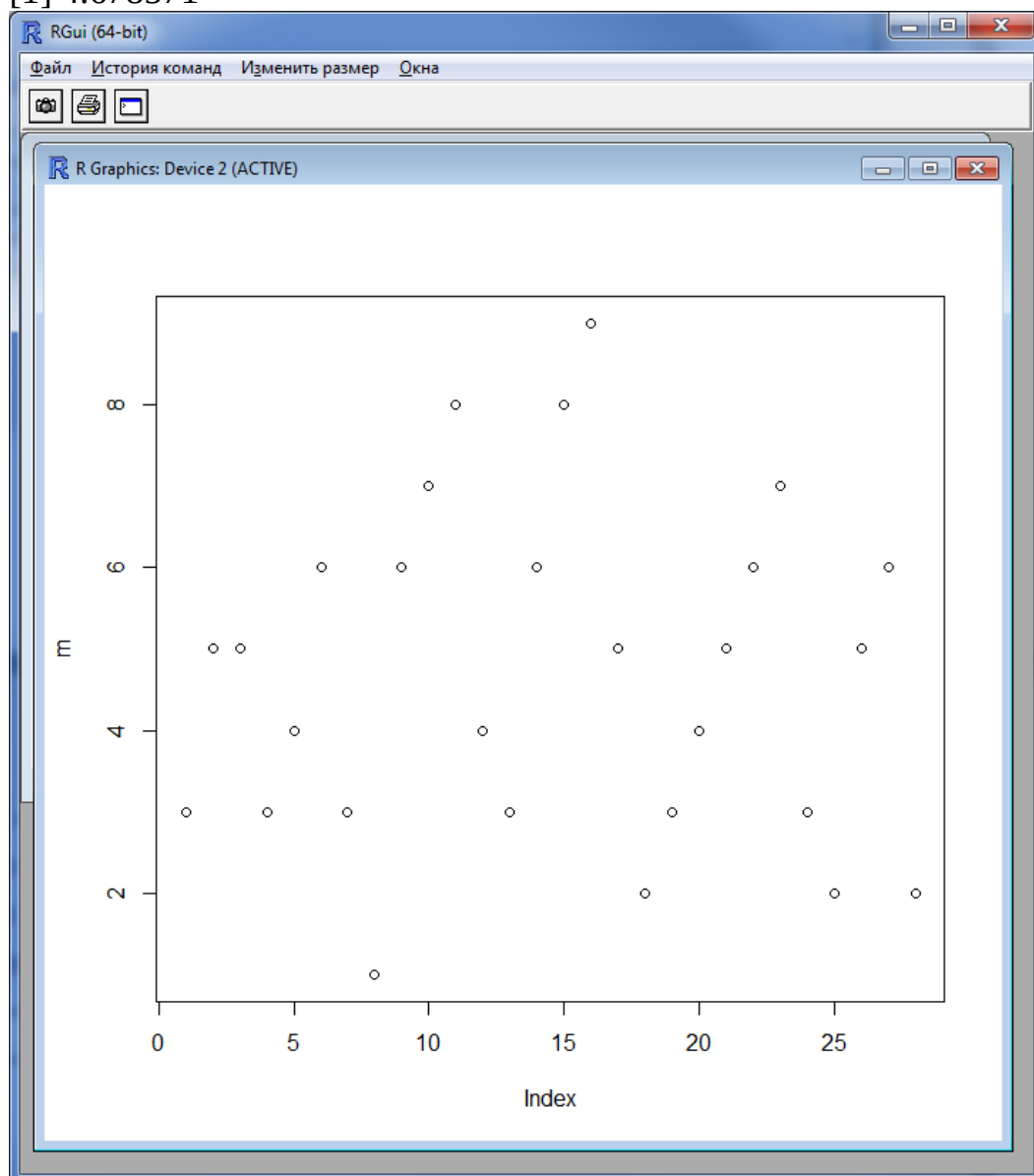
```
> val <- mean(m)
```

```
> val
```

```
> plot(m)
```

Результат:

```
[1] 4.678571
```

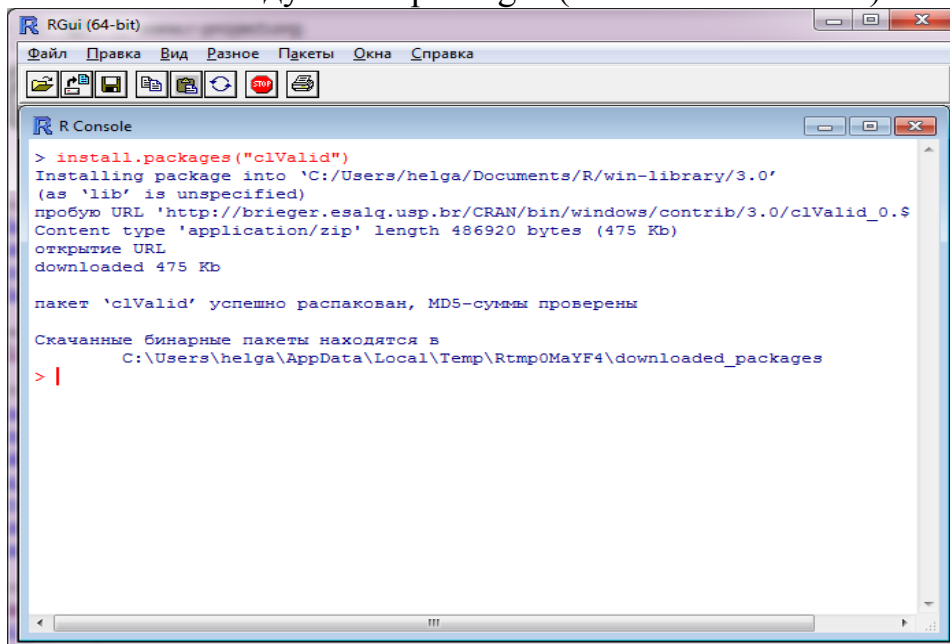


Приложение Г

Установка нового пакета для среды разработки.

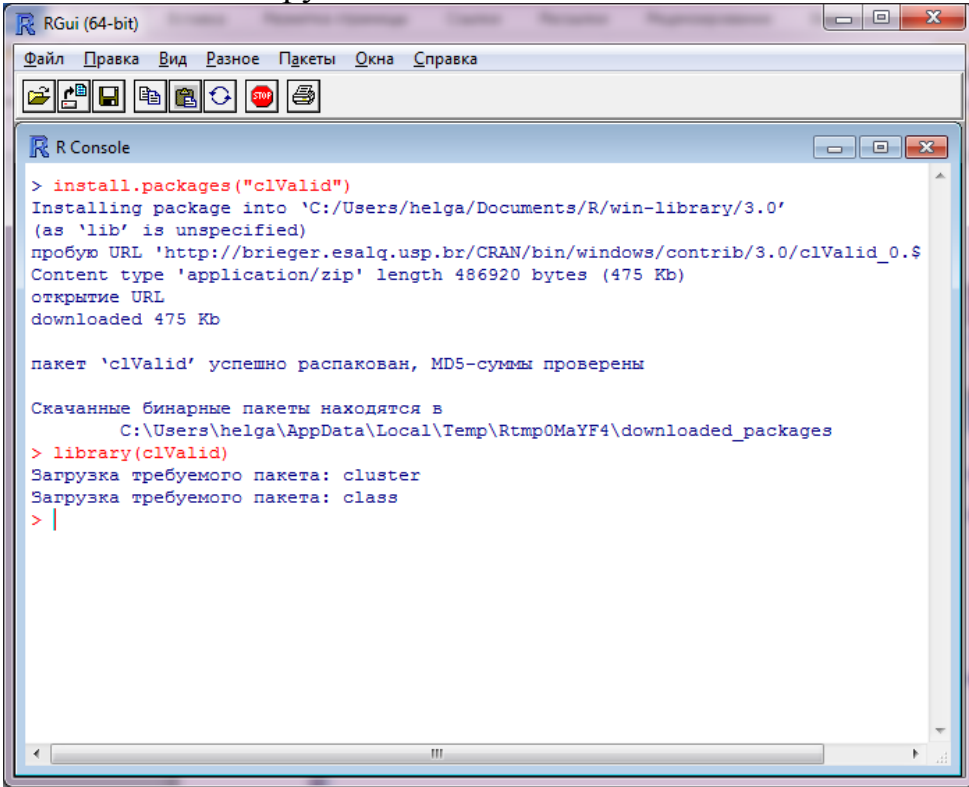
Для установки нового пакета для разработки программ следует воспользоваться командой `install.packages` («Название пакета»).

1. Открыть IDE.
2. Ввести команду `install.packages` («Название пакета») и нажать Enter.



3. Если появится окно с выбором откуда скачивать пакет, то выбрать зеркало для скачивания установок желательно из страны, которая находится ближе к России.

4. Для того чтоб преступить к использованию пакета следует ввести команду `library` (Название пакета). Обратите внимание, что название пакета не заключено в кавычки. IDE сама подгрузит необходимые пакеты, если они до этого не были загружены.



```
> install.packages("clValid")
Installing package into 'C:/Users/helga/Documents/R/win-library/3.0'
(as 'lib' is unspecified)
пробую URL 'http://brieger.esalq.usp.br/CRAN/bin/windows/contrib/3.0/clValid_0.$
Content type 'application/zip' length 486920 bytes (475 Kb)
открытие URL
downloaded 475 Kb

пакет 'clValid' успешно распакован, MD5-суммы проверены

Скачанные бинарные пакеты находятся в
C:\Users\helga\AppData\Local\Temp\Rtmp0MaYF4\downloaded_packages
> library(clValid)
Загрузка требуемого пакета: cluster
Загрузка требуемого пакета: class
> |
```

Приложение Д

Справочная информация по некоторым базовым функциям языка программирования R.

`help(topic)` — справка про `topic`
`help.search(«pattern»)` — глобальный поиск `pattern`
`help(package = )` — справка о заданном пакете
`help.start()` — запустить помощь в браузере
`apropos(what)` — имена объектов, которые соответствуют `what`
`args(name)` — аргументы команды `name`
`example(topic)` — примеры использования `topic`
`summary(object)` — общая информация об объекте `object`
`library(package)` — подключить пакет `package`
`array(data, dims)` — создаёт из `data` многомерные массив размерностей `dim`
`matrix(data, nrow = , ncol = , byrow = )` — создаёт из `data` матрицу `nrow` на `ncol`, порядок заполнения определяется `byrow`
`as.array(x)`, `as.data.frame(x)`, `as.numeric(x)`, `as.logical(x)`, `as.complex(x)`,
`as.character(x)` — преобразование переменной к заданному типу
`is.na(x)`, `is.null(x)`, `is.array(x)`, `is.data.frame(x)`, `is.numeric(x)`, `is.complex(x)`,
`is.character(x)` — проверка на то, что данный объект обладает указанным типом
`length(x)` — число элементов в `x`
`nrow(x)` — число строк `x`
`ncol(x)` — число столбцов `x`
`rev(x)` — реверсирует порядок элементов
`match(x, y)` — ищет элементы `x`, которые есть в `y`
`which(x == a)` — возвращает порядковые элементы `x`, которые равны `a`
`sin(x)`, `cos(x)`, `tan(x)`, `asin(x)`, `acos(x)`, `atan(x)`, `atan2(y, x)`, `log(x)`, `log(x, base)`,
`log10(x)`, `exp(x)` — элементарные математические функции
`min(x)`, `max(x)` — минимальный и максимальный элементы объекта
`sum(x)` — сумма элементов объекта
`prod(x)` — произведение элементов объекта
`diff(x)` — возвращает вектор из разниц между соседними элементами
`mean(x)` — среднее арифметическое элементов объекта
`median(x)` — медиана (средний элемент) объекта
`round(x, n)` — округляет `x` до `n` знаков после запятой
`fft(x)`, `mvfft(x)` — быстрое преобразование Фурье
`choose(n, k)` — количество сочетаний
`rank(x)` — ранжирует элементы объекта
`t(x)` — транспонированная матрица
`diag(x)` — диагональ матрицы
`print(x)` — выводит на экран
`plot(x, y)` — график зависимости `y` от `x`
`hist(x)` — гистограмма
`barplot(x)` — столбчатая диаграмма
`dotchart(x)` — диаграмма Кливленда

`pie(x)` — круговая диаграмма
`boxplot(x)` — график типа «коробочки»
`matplot(x, y)` — график зависимости столбцов `y` от столбцов `x`
`qqnorm(x)` — квантили
`qqplot(x, y)` — график зависимости квантилей `y` от квантилей `x`
`persp(x, y, z)` — то же, что и `image()`, но в виде трехмерного графика
`lines(x, y)` — рисование линии
`polygon(x, y)` — рисование многоугольника

Работа с функциями:

`function(arglist) { expr }` — создание пользовательской функции

`return(value)` — возвращение значения

`do.call(funname, args)` — вызывает функцию по имени

Условные операторы:

`if(cond) expr`

`if(cond) cons.expr else alt.expr`

`ifelse(test, yes, no)`

Циклы:

`for(var in seq) expr`

`while(cond) expr`

`repeat expr`

`break` — остановка цикла

`next` — переход к следующей итерации