

Министерство науки и высшего образования Российской Федерации  
Севастопольский государственный университет  
Кафедра ИС

Отчет  
по лабораторной работе №1  
«Исследование возможностей языка R для статического анализа данных»  
по дисциплине  
«ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ»

Выполнил студент группы ИС/б-17-2-о  
Горбенко К. Н.  
Проверил  
Сырых О.А.

Севастополь  
2020

## 1 ЦЕЛЬ РАБОТЫ

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

## 2 ЗАДАНИЕ НА РАБОТУ

1. Установить R на ПК.
2. Установить RStudio - инсталлятор скачать с официального сайта проекта.
3. Ознакомиться с кратким руководством пользователя RStudio.
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.

## 3 ХОД РАБОТЫ

Используем консоль R.exe. При входе получаем следующий текст:

```
1 R version 3.6.2 (2019-12-12) -- "Dark and Stormy"
2 Copyright (C) 2019 The R Foundation for Statistical Computing
3 Platform: x86_64-w64-mingw32/x64 (64-bit)
4
5 R is free software and comes with ABSOLUTELY NO WARRANTY.
6 You are welcome to redistribute it under certain conditions.
7 Type 'license()' or 'licence()' for distribution details.
8
9   Natural language support but running in an English locale
10 'citation()' on how to cite R or R packages in publications.
11
12 Type 'demo()' for some demos, 'help()' for on-line help, or
13 'help.start()' for an HTML browser interface to help.
14 Type 'q()' to quit R.
```

Выполним команду demo(). Результат выполнения команды представлен на рисунке 1.

```

R demos
File Edit

Demos in package 'base':

error.catching      More examples on catching and handling errors
is.things            Explore some properties of R objects and
                     is.FOO() functions. Not for newbies!
recursion            Using recursion for adaptive integration
scoping              An illustration of lexical scoping.

Demos in package 'graphics':

Hershey              Tables of the characters in the Hershey vector
                     fonts
Japanese             Tables of the Japanese characters in the
                     Hershey vector fonts
graphics             A show of some of R's graphics capabilities
image               The image-like graphics builtins of R
persp               Extended persp() examples
plotmath             Examples of the use of mathematics annotation

Demos in package 'grDevices':

colors               A show of R's predefined colors()
hclColors            Exploration of hcl() space

Demos in package 'stats':

```

Рисунок 1 – Результат выполнения команды `demo()`

Выполним команду `help(demo)`. Результат ее выполнения изображен на рисунке 2

```

demo {utils}
R Documentation

Demonstrations of R Functionality

Description
demo is a user-friendly interface to running some demonstration R scripts. demo() gives the list of available topics.

Usage
demo(topic, package = NULL, lib.loc = NULL,
      character.only = FALSE, verbose = getOption("verbose"),
      echo = TRUE, ask = getOption("demo.ask"),
      encoding = getOption("encoding"))

Arguments
topic
the topic which should be demonstrated, given as a name or literal character string, or a character
string, depending on whether character.only is FALSE (default) or TRUE. If omitted, the list of
available topics is displayed.

package
a character vector giving the packages to look into for demos, or NULL. By default, all packages in
the search path are used.

```

Рисунок 2 – Результат выполнения команды `help(demo)`

Работа с векторами:

- `c()` – создать вектор небольшой длины;
- `scan()` – считывание последовательно вводимые с клавиатуры значения;
- `mean()` – вычисляет среднее значение элементов вектора;
- `paste()` – объединяет элементы множества текстовых векторов;
- `sort()` – сортировка элементов вектора по возрастанию или убыванию;
- `class()` – проверка типа вектора;
- `sd()` – стандартное отклонение.

```

1 > v1 <- c(1, 2, 3, 4, 5) * 3
2 > v1
3 [1] 3 6 9 12 15
4 > v1[1:3]
5 [1] 3 6 9
6 > v2 = rep(15, 4)
7 > v2
8 [1] 15 15 15 15
9 > length(v2)
10 [1] 4

```

#### Работа с матрицами:

- `%*%` – матричное умножение;
- `t(x)` – транспонировать матрицу;
- `diag(x)` – диагональ матрицы;
- `solve(a, b)` – решает систему уравнений;
- `solve(a)` – обратная матрица;
- `colSums`, `rowSums`, `colMeans`, `rowMeans` – сумма и средние по столбцам и строкам.

```

1 > mat1 <- matrix (data=1, nrow=3, ncol=3)
2 > mat1
3      [,1] [,2] [,3]
4 [1,]    1    1    1
5 [2,]    1    1    1
6 [3,]    1    1    1
7 > is.matrix(mat1)
8 [1] TRUE
9 > dim(mat1)
10 [1] 3 3
11 > dim(mat1)
12 [1] 3 3

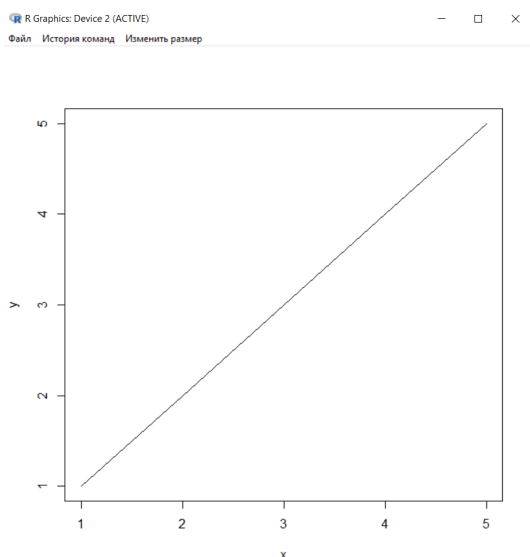
```

#### Работа с графикой:

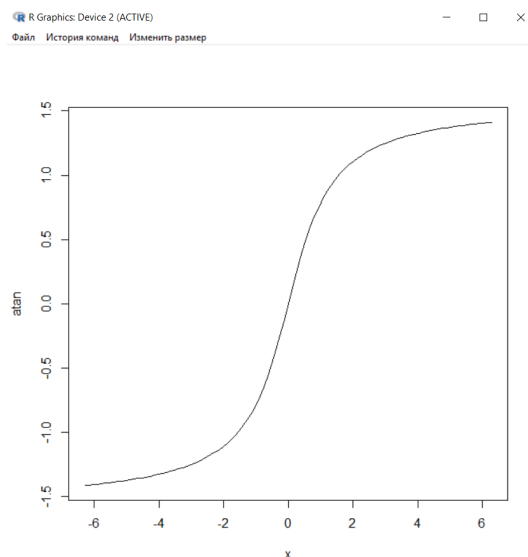
- `matplot(x,y)` – график зависимости столбцов  $y$  от столбцов  $x$ ;
- `foutfoldplot(x)` – изображает (в виде частей окружности) связь между двумя бинарными переменными в разных совокупностях;
- `assocplot(x)` – график Кохена-Френдли;
- `pairs(x)` – функция изображает диаграммы рассеяния для всех возможных пар переменных из матрицы  $x$ ;
- `plot.ts(x)`, `ts.plot(x)` – изображает временной ряд;
- `qqnorm(x)` – квантили;
- `qqplot(x, y)` – график зависимости квантилей  $y$  от квантилей  $x$ ;
- `contour(x, y, z)` – выполняет интерполяцию данных и создает контурный график;
- `symbols(x, y)` – изображает различные символы в соответствии с координатами;
- `termplot(mod.obj)` – изображает частные эффекты переменных из регрессионной модели;

```
1 > x <- c(1, 2, 3, 4, 5)
2 > y <- c(1, 2, 3, 4, 5)
3 > plot(x, y, type="l")
4 > plot(atan, -2 * pi, 2 * pi)
```

Результат выполнения программы:



(a)  $y = x$



(b)  $y = \arctan(x)$

Рисунок 3 – Демонстрация работы с графикой

## 4 КОНТРОЛЬНЫЕ ВОПРОСЫ

### 4.1 Особенности языка R

Язык R — язык программирования для статистической обработки данных и работы с графикой, а также свободная программная среда вычислений с открытым исходным кодом.

R — интерпретируемый язык программирования, основным способом работы с которым является командный интерпретатор. Язык объектный: любой программный объект в нём имеет набор атрибутов — именованный список значений, определяющих его.

Язык поддерживает минимальный набор примитивных типов данных: символьный, числовой, логический и комплексный. Числовые переменные, помимо обычных чисел, могут принимать специальные значения NaN (Not a Number — «не число») и Inf (Infinity — «бесконечность»). Бесконечность (положительная или отрицательная) получается при выходе результата вычислений за пределы представимого реализацией диапазона, NaN — при операциях с неопределённым результатом. Помимо этих, имеется ещё одно очень важное специальное значение, NA (Not Available — «не доступно»). Оно может быть использовано для фиксации того факта, что соответствующее значение, участвующее в вычислениях, по какой-либо причине не было.

Значения примитивных типов могут объединяться в векторы (vector), списки (list), матрицы или массивы (matrix), в том числе многомерные; эти комбинированные типы хранят наборы данных одного и того же примитивного типа. Помимо этого, язык содержит понятие факторов (factor) — наборов категориальных или шкальных данных, принимающих строго определённый набор значений. Наконец, могут создаваться таблицы (data frame) — структуры данных, которые для каждой строки (индивида) хранят набор различных (и имеющих разные типы) параметров (признаков). Особенностью R является то, что операции с векторами и матрицами поддерживаются на уровне самого языка.

Существует операция извлечения и записи данных (аналог присваивания) “<-“, а также обычные операции работы с данными, в том числе арифметические. Доступ по индексу к элементам векторов и массивов осуществляется с помощью квадратных скобок (отсчет производится с 1), доступ к атрибутам списков

— посредством оператора «\$». Имеется минимальный набор обычных конструкций императивного программирования: условный оператор if, циклы while и for. Выражения на R можно описывать как отдельные объекты и вычислять по мере необходимости. На этом же механизме основано описание функций.

Функции R могут объединяться в пакеты — загружаемые модули, которые подключаются к любой программе и предоставляют объединённые в них вычислительные средства. Пакеты для R могут разрабатываться на других языках программирования, в том числе на Си.

Сам язык имеет довольно ограниченные и не слишком удобные средства описания данных, но это компенсируется наличием библиотечных средств, которые позволяют загружать в виде таблиц R наборы данных, представленных в большинстве открытых и многих проприетарных форматах. Так, в R могут быть легко загружены таблицы в простом текстовом формате, таблицы Excel различных версий, данные в форматах CSV, XML и многих других.

В R реализованы практически все актуальные средства универсальных статистических вычислений, такие как регрессионный анализ и анализ временных рядов, а также множество специфических алгоритмов для решения узкоспециализированных задач и исследований в отдельных областях.

## **4.2 Команда для получения подробной информации о функции в R**

Для получения подробной информации (справки) о любой функции, необходимо выполнить команду:

`help(имяФункции) (?имяФункции).`

## **4.3 Структура и особенности команды round() в R**

`round(x, digits = 0)` - функция для округления значения(значений) первого параметра до количества символов после запятой, задаваемого вторым параметром (0 по умолчанию).

Из-за различной реализации стандарта IEEE 754 на разных системах поведение `round()` при округлении символа 5 может отличаться (округляет в большую либо меньшую сторону)

Отрицательный параметр количества символов обозначает округление до

степени 10.

#### 4.4 Команды для работы с векторами в R

- `c()` – создать вектор небольшой длины;
- `scan()` – считывание последовательно вводимые с клавиатуры значения;
- `mean()` – вычисляет среднее значение элементов вектора;
- `paste()` – объединяет элементы множества текстовых векторов;
- `sort()` – сортировка элементов вектора по возрастанию или убыванию;
- `class()` – проверка типа вектора;
- `sd()` – стандартное отклонение.

#### 4.5 Команды для работы с матрицами в R

- `%*%` – матричное умножение;
- `t(x)` – транспонировать матрицу;
- `diag(x)` – диагональ матрицы;
- `solve(a, b)` – решает систему уравнений;
- `solve(a)` – обратная матрица;
- `colSums`, `rowSums`, `colMeans`, `rowMeans` – сумма и средние по столбцам и строкам.

#### 4.6 Работа с графикой в R

- `matplot(x,y)` – график зависимости столбцов `y` от столбцов `x`;
- `foutfoldplot(x)` – изображает (в виде частей окружности) связь между двумя бинарными переменными в разных совокупностях;
- `assocplot(x)` – график Кохена-Френдли;
- `pairs(x)` – функция изображает диаграммы рассеяния для всех возможных пар переменных из матрицы `x`;
- `plot.ts(x)`, `ts.plot(x)` – изображает временной ряд;
- `qqnorm(x)` – квантили;
- `qqplot(x, y)` – график зависимости квантилей `y` от квантилей `x`;
- `contour(x, y, z)` – выполняет интерполяцию данных и создает контурный график;
- `symbols(x, y)` – изображает различные символы в соответствии с коорди-



натами;

- `termplot(mod.obj)` – изображает частные эффекты переменных из регрессионной модели;

## **ВЫВОДЫ**

В ходе лабораторной работы было получено начальное представление об особенностях языка (платформы R): о запуске интерпретатора, об операторах, об операциях над векторами, матрицами, о работе с графикой.