

**Н.Ю. ДОБРОВОЛЬСКАЯ  
О.В. ГАРКУША**

# **Язык программирования R**

**Практикум**



**Краснодар  
2024**

Министерство науки и высшего образования Российской Федерации  
КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Н.Ю. ДОБРОВОЛЬСКАЯ  
О.В. ГАРКУША

# ЯЗЫК ПРОГРАММИРОВАНИЯ R

Практикум

Краснодар  
2024

УДК 004.43(075.8)

ББК 32.973.2 я 73

Д 56

Рецензенты:

Доктор физико-математических наук, профессор

*Е.Н. Калайдин*

Кандидат физико-математических наук, доцент

*С.Е. Рубцов*

**Добровольская, Н.Ю., Гаркуша, О.В.**

Д 56      Язык программирования      R:      практикум /  
Н.Ю. Добровольская, О.В. Гаркуша; Министерство науки и  
высшего образования Российской Федерации, Кубанский  
государственный университет. – Краснодар: Кубанский гос.  
ун-т, 2024. – 80 с. – 500 экз.  
ISBN 978-5-8209-2492-7

Предлагаемое издание содержит теоретические положения, задания и методические указания к выполнению лабораторных работ по дисциплине «Язык программирования R».

Адресуется студентам бакалавриата направлений подготовки 09.03.03 «Прикладная информатика» профиль «Прикладная информатика в экономике».

УДК 004.43(075.8)

ББК 32.973.2 я 73

ISBN 978-5-8209-2492-7

© Кубанский государственный  
университет, 2024

© Добровольская Н.Ю.,  
Гаркуша О.В., 2024

# **1. ОРГАНИЗАЦИЯ ЗАНЯТИЙ ПО КУРСУ**

Дисциплина «Язык программирования R» читается студентам направления 09.03.03 «Прикладная информатика» профиль «Прикладная информатика в экономике» на III курсе.

Цель курса – формирование у студентов знаний, умений и практических навыков в области анализа данных, математической и статистической обработки числовых данных, визуализации данных.

В результате выполнения лабораторной работы оформляется отчет. Отчеты сдаются на проверку руководителю по мере их выполнения и защищаются студентами в установленном порядке.

При защите отчета студенту могут быть заданы вопросы и дополнительные задания по сути лабораторной работы. Кроме того, студенты самостоятельно выполняют шесть индивидуальных заданий (в соответствии с номером варианта), решения которых добавляются в отчет к текущей лабораторной работе. При неудовлетворительной оценке знаний по теме данного отчета студент возвращается к повторному изучению соответствующих материалов, после чего допускается к повторной защите. Неудовлетворительно выполненный отчет также возвращается на доработку.

## **1.1. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ ОТЧЕТОВ О ВЫПОЛНЕНИИ ЛАБОРАТОРНЫХ РАБОТ**

В процессе изучения курса «Язык программирования R» должны быть подготовлены и сданы восемь отчетов о выполнении лабораторных работ (см. таблицу). Лабораторные работы выполняются в соответствии с индивидуальной тематикой (см. приложение).

Отчет должен содержать заголовок, тему лабораторной работы, цель, задание, индивидуальную тему, описание хода выполнения работы, фрагменты кода на языке программирования R и результаты их выполнения (допускаются скриншоты), в соответствии с требованиями к содержанию и выводы по работе.

## 1.2. ГРАФИК ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ

№	Тема лабораторной работы: материал для изучения и характер выполняемых работ	Неделя выполнения
1	Обработка целых чисел	1–2
2	Генерация последовательности чисел. Стандартные операции с вектором	3–4
3	Операторы языка R. Обработка векторов. Индивидуальное задание 1	5–6
4	Использование математических функций. Индивидуальное задание 2	7–8
5	Матрицы. Индивидуальное задание 3	9–10
6	Функции. Индивидуальное задание 4	11–12
7	Вычисление интегралов и производных. Таблицы. Индивидуальное задание 5	13–14
8	Датасеты. Исследование характеристик объектов индивидуальной тематики. Индивидуальное задание 6	15–16

## 1.3. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ОТЧЕТНОЙ ДОКУМЕНТАЦИИ

Отчет должен быть выполнен электронным способом с использованием компьютера. Для создания текста работы рекомендуется использовать текстовый редактор MS Word. Шрифт на протяжении всего документа должен быть одинаковый: Times New Roman, кегль 14, за исключением оформления иллюстраций, таблиц и формул, в которых допускается использовать шрифт меньшего размера. Минимально допустимый размер шрифта – 12.

При оформлении текста работы следует использовать абзацный отступ.

**Нумерация страниц отчета.** Страницы отчета следует нумеровать арабскими цифрами, соблюдая сквозную нумерацию по всему тексту. Номер страницы проставляют в центре нижней части листа без точки. Титульный лист включают в общую нумерацию страниц отчета. Номер страницы на титульном листе не проставляют. Иллюстрации и таблицы, расположенные на отдельных листах, включают в общую нумерацию страниц отчета.

**Иллюстрации.** Иллюстрации (скриншоты) следует располагать в отчете непосредственно после текста, в котором они

упоминаются впервые, или на следующей странице. Иллюстрации могут быть в компьютерном исполнении, в том числе и цветные. На все иллюстрации должны быть даны ссылки в отчете.

Иллюстрации следует нумеровать арабскими цифрами сквозной нумерацией. Если рисунок один, то он обозначается «Рисунок 1». Слово «Рисунок» и его наименование располагают посередине строки. Допускается нумеровать иллюстрации в пределах раздела. В этом случае номер иллюстрации состоит из номера раздела и порядкового номера иллюстрации, разделенных точкой. Например, Рисунок 1.1. Иллюстрации, при необходимости, могут иметь наименование и пояснительные данные (подрисуночный текст). Слово «Рисунок» и наименование помещают после пояснительных данных и располагают следующим образом: «Рисунок 1.1 — Функциональная схема».

При ссылках на иллюстрации следует писать «... в соответствии с рисунком 2» при сквозной нумерации и «... в соответствии с рисунком 1.2» при нумерации в пределах раздела.

## 2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ

### ЛАБОРАТОРНАЯ РАБОТА №1

#### *Обработка целых чисел*

Запустить среду RStudio. Используя арифметические операции языка, R выполнить задание.

#### *Арифметические операции:*

+ (сложение);  
– (вычитание);  
\* (умножение);  
/ (деление);  
^ (возведение в степень);  
%% (целочисленное деление);  
%% (остаток от деления).

#### *Условный оператор*

Краткая форма:

if (условие) выражение

Полная форма:

if (условие) выражение else выражение

условие – любой оператор условия (==, <=, >=, >, <, !=), результатом выполнения которого является логический вектор единичной длины. Если значение вектора TRUE, то выполняется выражение.

выражение – один или несколько операторов. Если операторов несколько, они разделяются точкой с запятой «;» и заключаются в фигурные скобки «{ }».

Сформировать векторы присваиванием, функциями seq() и c().

c(...) – объединяет аргументы в вектор, например c(1, 2, 3)

seq(from, to, by = ) – генерирует последовательность чисел от from до to с шагом by;

seq(from, to, len = ) – генерирует последовательность чисел от from до to длины len.

### Пример 1. Генерация векторов.

```
> seq(10) # Apply seq function
[1] 1 2 3 4 5 6 7 8 9 10

> seq(3,10) # seq function from to
[1] 3 4 5 6 7 8 9 10

> seq(3,10,0.5) # seq function by 0.5
[1] 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5
[11] 8.0 8.5 9.0 9.5 10.0

> seq(3, 10, length.out = 5) # Sequence of length 5
[1] 3.00 4.75 6.50 8.25 10.00
```

Для вектора вычислить следующие значения функций:

sum(x)	Сумма элементов объекта x.
prod(x)	Произведение элементов объекта x.
max(x)	Максимальное значение из объекта x.
min(x)	Минимальное значение из объекта x.

**Пример 2.** Вычисление функций для целочисленного вектора.

```
> x = seq(3, 50, 3)
> prod(x)
[1] 9.006575e+20

> x = seq(50, 10, -3)
> sum(x)
[1] 427

> x = c(50, 16, 78, -90, 10, -3)
> max(x)
[1] 78
> min(x)
[1] -90
```



**Пример 3.** Сгенерировать целочисленный вектор. Если его второй элемент оканчивается на 8, то найти наибольшее значение вектора, в противном случае – наименьшее значение. Нумерация вектора начинается с единицы.

```
> x = c(50, 18, 78, -90, 10, -3)
> if (x[2] %% 10==8) print(max(x)) else print(min(x))
```

```
[1] 78
```

```
> x = c(50, 58, 78, -90, 10, -3)
> k = 0
> if (x[1] %% 10 == 5) k = k + 1
> if (x[2] %% 10 == 5) k = k + 1
> print (k)
```

```
[1] 2
```

**Пример 4.** Дано пятизначное число. Верно ли, что его первая и последняя цифры совпадают?

```
> x = 12345
> if (x %% 10 == x %% 10000) print ("Yes") else
+   print ("No")
```

```
[1] "No"
```

**Пример 5.** Дано два числа. Если они разных знаков, то найти сумму чисел, в противном случае найти произведение чисел.

```
> a = 150
> b = -200
> if (a>0 && b<0 || a<0 && b>0) print(a + b) else
+   print(a * b)
```

```
[1] -50
```

### *Задания*

1. Используя операции целочисленного деления  $\%/\%$  и остатка  $\%\%$  для случайного четырехзначного числа, вывести отдельно его цифры.

2. Дано случайное трехзначное целое число. Найти произведение его цифр.

3. Дано четырехзначное число, поменять местами его первую и последнюю цифры.

4. Дано пятизначное число. Найти произведение его первой и третьей цифр.

5. Дано пятизначное целое положительное число. Сформировать новое число – в обратном порядке следования цифр.

6. Дано четырехзначное число. Определите, что больше: сумма крайних цифр или квадрат третьей цифры?

7. Дано шестизначное число. Определите, является ли оно палиндромом.

8. Дано два целых числа. Если они разных знаков и различной четности, то вывести сумму их квадратов. Если только одно число делится на 5, то вывести сумму их кубов. Если хотя бы одно число оканчивается на 111, то увеличить оба числа вдвое.

9. Сгенерировать несколько различных векторов (перечисление элементов, генерация значений с шагом, генерация значений заданной длины). Найти сумму, произведение, наибольший и наименьший элементы полученных векторов.

## ЛАБОРАТОРНАЯ РАБОТА №2

### *Генерация последовательности чисел. Стандартные операции с вектором*

Запустить среду RStudio. Сформировать несколько векторов с помощью явной инициализации, функций `seq()` и `c()`.

Для векторов вычислить следующие значения функций:

Имя функции	Действие / результат
<code>which.max(x)</code>	Индекс максимального значения объекта <code>x</code> .
<code>which.min(x)</code>	Индекс минимального значения объекта <code>x</code> .
<code>range(x)</code>	Минимальное и максимальное значения объекта <code>x</code> .
<code>length(x)</code>	Число элементов в объекте <code>x</code> .
<code>mean(x)</code>	Среднее значение элементов объекта <code>x</code> .
<code>median(x)</code>	Медиана объекта <code>x</code> .
<code>round(x,n)</code>	Округляет элементы <code>x</code> до <code>n</code> знаков после запятой.
<code>rev(x)</code>	Перестановка элементов <code>x</code> в обратном порядке.
<code>sort(x)</code>	Сортирует элементы <code>x</code> в возрастающем порядке.
<code>rev(sort(x))</code>	Сортирует в убывающем порядке.
<code>rank(x)</code>	Выдает классы элементов <code>x</code> .
<code>log(x, base)</code>	Вычисляет логарифм <code>x</code> с основанием <code>base</code> .
<code>pmin(x,y,...)</code>	Вектор, в котором <code>i</code> -й элемент минимальный из значений <code>x[i]</code> , <code>y[i]</code> ...
<code>pmax (x, y...)</code>	Вектор, в котором <code>i</code> -й элемент максимальный из значений <code>x[i]</code> , <code>y[i]</code> ...
<code>cumsum(x)</code>	Вектор, <code>i</code> -й элемент которого является суммой значений от <code>x[1]</code> до <code>x[i]</code>
<code>cumprod (x)</code>	Вектор, <code>i</code> -й элемент которого является произведением значений от <code>x[1]</code> до <code>x[i]</code> .
<code>cummin (x)</code>	Вектор, <code>i</code> -й элемент которого является минимальным из элементов от <code>x[1]</code> до <code>x[i]</code> .
<code>cummax (x)</code>	Вектор, <code>i</code> -й элемент которого является максимальным из элементов от <code>x[1]</code> до <code>x[i]</code> .

### *Генерация случайных чисел*

Для генерации целых чисел в диапазоне  $[min, max]$  используется функция

`sample(x, size, replace = FALSE, prob = NULL)`,

которая в качестве аргументов принимает:

`x` – целое число (эквивалентно `1:x`) или вектор, ограничивающий диапазон (например, `5:15`);

`size` – количество генерируемых значений;  
`replace` – разрешаются ли повторения;  
`prob` – вектор весов распределения, по умолчанию отсутствует (равномерное распределение.)

```
> x3 <- sample(1:10, 1)
> x3
[1] 8
> x4 <- sample(1:10, 5, replace=T)
> x4
[1] 10 3 8 8 8
> x5 <- sample(1:10, 5, replace=F)
> x5
[1] 2 3 1 8 6
```

Для генерации вещественных чисел в диапазоне  $[min, max]$  используется функция

`runif(n, min = 0, max = 1),`

которая в качестве аргументов принимает:

`n` – количество генерируемых значений;

`min` – нижняя граница диапазона; вещественное конечное число;

`max` – верхняя граница диапазона; вещественное конечное число.

```
> x1 <- runif(1, 5.0, 7.5)
> x1
[1] 6.996858
> x2 <- runif(10, 5.0, 7.5)
> x2
[1] 6.252209 5.594898 5.606375 6.374769 5.957075
[6] 7.022946 5.562264 7.184194 6.843230 6.555839
```

**Пример 1.** Сгенерировать вектор целых случайных чисел размерности 5. Найти сумму элементов вектора.

```
> x<-sample(1:10)
> x
[1] 8 7 5 6 1 9 10 3 4 2

> sum(x)
[1] 55
```

**Пример 2.** Сгенерировать вектор вещественных случайных чисел размерности 10. Получить новый вектор, каждый элемент которого является суммой элементов исходного вектора от начала до текущего элемента.

```
> x<-runif(10,5.0,10.5)
> x
[1] 6.785055 10.110786 6.512287 9.497474 6.662449
[6] 9.095209 10.179797 6.080152 5.598472 9.294501

> cumsum(x)
[1] 6.785055 16.895841 23.408128 32.905602 39.568051
[6] 48.663260 58.843057 64.923209 70.521681 79.816182
```

**Пример 3.** Сгенерировать вектор целых чисел. Если его первый элемент нечетный, то вывести элементы вектора в возрастающем порядке, в противном случае – в убывающем порядке.

```
> x = c(90, 76, -56, 93, -76)
> if (x[1] %% 2 != 0) print(sort(x)) else
+   print(rev(sort(x)))
[1] 93 90 76 -56 -76
```

**Пример 4.** Сгенерировать вектор целых чисел. Если сумма элементов вектора кратна трем, то вывести второй элемент кумулятивной суммы, в противном случае – вывести третий элемент кумулятивного произведения.

```
> x = c(-57, 97, -34, 125, -765, 6)
> if (sum(x) %% 3 ==0) print(cumsum(x)[2]) else
+   print(cumprod(x)[3])
[1] 187986
```

### Задания

1. Сгенерировать вектор в соответствии с индивидуальной тематикой (см. приложение).

Например, имеется задание – организовать продажу билетов на рейсы. Журнал продаж билетов: стоимость билета, тип билета, длительность маршрута в минутах, суммарные продажи по дням.

Необходимо сгенерировать векторы целых случайных чисел размерности 50 (с помощью функций R) и сохранить в таблице (далее «Таблица исходных данных») для дальнейшей работы (файл .xls или .xlsx).

Таблица исходных данных

№	Стоимость билета	Тип билета	Длительность маршрута в минутах	Дата
1	2300	A	540	7.01
2	5480	A	370	7.01
3	3489	Э	1290	8.01

2. Выполнить задание на сгенерированном наборе данных для некоторого столбца Таблицы исходных данных:

- найти произведение элементов вектора;
- найти наибольшее значение;
- найти номер наименьшего элемента;
- найти медиану вектора;
- получить новый вектор – перестановку элементов в обратном порядке исходного вектора;
- получить новый вектор, каждый элемент которого равен наибольшему элементу от начала вектора до текущего элемента.

3. Выполнить задание на сгенерированном наборе данных для некоторого столбца Таблицы исходных данных:

- найти наименьший элемент;
- найти индекс наибольшего элемента;
- найти среднее арифметическое элементов и округлить до 2 знаков;
- выполнить кумулятивное сложение и умножение на первых пяти элементах;
- отсортировать вектор в порядке убывания и вывести значения с округлением до двух знаков после запятой;

- построить новый вектор, каждый элемент которого является произведением элементов от начала до текущего элемента. Вывести новый вектор с округлением до одного знака после запятой.

4. Выполнить задание на сгенерированном наборе данных для двух столбцов Таблицы исходных данных: найти среднее арифметическое суммы этих векторов, а затем вывести новый вектор – сумму исходных векторов в порядке убывания значений элементов.

## ЛАБОРАТОРНАЯ РАБОТА №3

### *Операторы языка R. Обработка векторов*

Условный оператор имеет вид. Краткая форма:  
if (условие) выражение

Полная форма:  
if (условие) выражение else выражение

Если выражение представлено более чем одной командой, необходимо заключить команды в {}.

```
>x=6
> y=123
> if (x==y%%10)  {
+   x=x+1
+ } else {x=x-1}
> if (x!=y/10%%10) {x=x+y; x}
[1] 128
```

### *Условная функция*

Функция ifelse (условие, yes, no) позволяет получить новое значение для переменной в зависимости от выполнения условия. Применяется к векторам.

**Пример 1.** Сгенерировать вектор случайных целых чисел. Получить новый вектор, значения которого – корень квадратный из значений исходного вектора.

Извлечение корня из отрицательного значения дает NaN, чтобы этого избежать, можно использовать условную функцию.

```
> x=sample(-100:100,10)
> z=sqrt(x)
Warning message:
In sqrt(x) : NaNs produced
> z
[1] 8.944272      NaN 2.236068      NaN      NaN      NaN
[7]      NaN 8.660254 4.472136      NaN
> z=ifelse(x>0,sqrt(x),x)
Warning message:
In sqrt(x) : NaNs produced
> z
```



```
[1] 8.944272 -28.000000 2.236068 -98.000000 -59.000000
[6] -73.000000 -52.000000 8.660254 4.472136 -93.000000
```

### ***Оператор цикла for***

for (переменная in последовательность) выражение

Пока переменная находится в рамках заданной числовой последовательности, выполняется выражение (тело цикла).

**Пример 2.** Дано два вектора. Получить новый вектор по заданному правилу.

```
> x=sample(1:100,10)
> x
[1] 5 95 75 100 45 40 74 86 77 29
> y=sample(1:100,10)
> y
[1] 65 43 35 10 4 7 96 79 13 74
> w=vector()
> for (i in 1:10)
+ { if (x[i]<y[i]) {w[i]=x[i]/y[i]} else {w[i]=x[i]*y[i]}}
> w
[1] 7.692308e-02 4.085000e+03 2.625000e+03 1.000000e+03
[5] 1.800000e+02 2.800000e+02 7.708333e-01 6.794000e+03
[9] 1.001000e+03 3.918919e-01
```

Если указывать только w, то будет формироваться не вектор, а переменная (выражение вычисляется только для первых значений векторов).

```
> x=sample(1:100,10)
> x
[1] 5 42 35 58 82 90 67 13 14 56
> y=sample(1:100,10)
> y
[1] 29 5 56 31 53 97 10 86 30 81
> w=vector()
> for (i in 1:10) {
+   if (x[i]<y[i]) {
+     w=x[i]/y[i]
+   } else {
+     w=x[i]*y[i]
+   }
}
```

```
+ }  
> w  
[1] 0.691358
```

**Пример 3.** Найти сумму четных элементов вектора.

```
> x=sample(1:100,10)  
> x  
[1] 53 69 55 23 96 59 8 21 78 56  
> s=0  
> for(i in 1:10)  
+   if (x[i]%%2==0) s=s+x[i]  
> s  
[1] 238
```

### *Оператор цикла while*

**while** (условие) выражение

Выражение будет вычисляться, пока значение условия True.

**Пример 4.** Найти сумму цифр натурального числа.

```
> x=sample(100:9999,1)  
> x  
[1] 5274  
> s=0  
> while (x>0)  
+ { s=s+x%%10; x=x%%10; s}  
> s  
[1] 18
```

При работе с векторами удобно использовать функцию, определяющую индекс некоторого элемента. Для этого применяется функция `which()`.

**Пример 5.** Пусть задан вектор `x`.

```
> x=c(10, 4, 7, 9, 7, 8, 6, 14, 10, 10, 5, 8, 11, 11, 20)  
> length(x)  
[1] 15
```

Определить номера вектора `X`, большие 10.

```
> which(x>10)
```

```
[1] 8 13 14 15
```

Если в векторе необходимо найти второй элемент, превосходящий 10, то достаточно указать

```
> x[13]
```

```
[1] 11
```

либо

```
> y=which(x>10)
```

```
> x[y[2]]
```

```
[1] 11
```

**Пример 6.** Найти в векторе X второй по порядку элемент, кратный 5.

```
> x
```

```
[1] 10 4 7 9 7 8 6 14 10 10 5 8 11 11 20
```

```
>
```

```
> y=which(x%%5==0);y
```

```
[1] 1 9 10 11 15
```

```
>
```

```
> x[y[2]]
```

```
[1] 10
```

Дополнительно можно использовать функции `which.max()` и `which.min()`, которые находят соответственно номер максимального и минимального элементов вектора.

```
> which.max(x)
```

```
[1] 15
```

```
> which.min(x)
```

```
[1] 2
```

**Пример 7.** Найти все элементы вектора, значение которых наиболее близко к 12.

```
> y=which(abs(x-12)==min(abs(x-12)));y
```

```
[1] 13 14
```

И вывести их на экран.

```
> x[y]
```

```
[1] 11 11
```

### *Задания*

1. Дано положительное четырехзначное число. Если оно четно, то вывести сумму его цифр. В противном случае определить, какая цифра больше: первая или последняя.
2. Используя данные из Таблицы исходных данных:
  - найти сумму положительных не кратных 5 элементов;
  - найти произведение порядковых номеров элементов с последней цифрой 5;
  - найти номер элемента, имеющего первое нечетное значение;
  - найти номер элемента, имеющего последнее значение, делящееся нацело на 7.
3. Используя данные из Таблицы исходных данных:
  - найти произведение элементов вектора, оканчивающихся на четную цифру;
  - найти сумму элементов вектора, расположенных между первым и последним нулевыми элементами.
4. Используя данные из Таблицы исходных данных:
  - найти сумму элементов вектора, сумма двух последних цифр у которых больше 10;
  - найти сумму элементов вектора, расположенных между первым и последним отрицательными элементами.
5. Дано случайное целое положительное число:
  - найти сумму четных цифр числа;
  - найти наибольшую цифру числа;
  - найти наименьшую нечетную цифру числа;
  - верно ли, что в числе нет цифр 2 и 3;
  - верно ли, что в числе все цифры четные;
  - верно ли, что в числе цифр 5 больше, чем четных цифр?
6. Используя данные из Таблицы исходных данных, выполнить сложение двух векторов и умножение на константу. С помощью команды `plot(x)` вывести элементы вектора графически.
7. Выполнить индивидуальное задание.

**Индивидуальное задание**  
**(обработка вектора)**

- 3.1.** а) найти максимальный элемент вектора, не оканчивающийся на цифру 7;  
б) найти сумму элементов вектора, расположенных до последнего положительного элемента.
- 3.2.** а) найти минимальный элемент вектора, оканчивающийся на цифру 4;  
б) найти сумму элементов вектора, расположенных между первым и последним положительными элементами.
- 3.3.** а) найти номер максимального элемента вектора, не содержащего цифру 5;  
б) найти произведение элементов вектора, расположенных между первым и вторым нулевыми элементами.
- 3.4.** а) найти номер минимального элемента вектора, делящегося на 11;  
б) найти сумму элементов вектора, расположенных между первым и вторым отрицательными элементами.
- 3.5.** а) найти максимальный элемент вектора, не оканчивающийся на цифры 2 и 3;  
б) найти сумму элементов вектора, расположенных между первым и вторым положительными элементами.
- 3.6.** а) найти сумму элементов вектора, произведение двух последних цифр которых больше 15;  
б) найти сумму модулей элементов вектора, расположенных после первого элемента, равного нулю.
- 3.7.** а) найти произведение элементов вектора, которые делятся на 3 или на 5;  
б) найти сумму модулей элементов вектора, расположенных после первого отрицательного элемента.
- 3.8.** а) найти номер первого подходящего элемента вектора, две последние цифры (двузначное число) которого больше значения первого элемента вектора;  
б) найти сумму элементов вектора, расположенных после первого четного элемента.
- 3.9.** а) найти количество элементов вектора, три последние цифры (трехзначное число) которых находятся в диапазоне  $[A, B]$ ;

- б) найти сумму элементов вектора, расположенных после последнего вхождения максимального элемента.
- 3.10.** а) найти количество элементов вектора, не оканчивающихся на ноль;  
б) найти сумму элементов вектора, расположенных после первого вхождения минимального элемента.
- 3.11.** а) найти количество элементов вектора, произведение двух последних цифр у которых больше  $C$ ;  
б) найти произведение элементов вектора, расположенных после максимального по модулю элемента.
- 3.12.** а) найти количество элементов вектора, не оканчивающихся на цифру 7;  
б) найти сумму модулей элементов вектора, расположенных после минимального по модулю элемента.
- 3.13.** а) найти количество элементов вектора, сумма трех последних цифр у которых нечетна;  
б) найти сумму элементов вектора, расположенных после последнего четного элемента.
- 3.14.** а) найти сумму элементов вектора, сумма двух последних цифр у которых меньше  $C$ ;  
б) найти сумму целых частей элементов вектора, расположенных после последнего отрицательного элемента.
- 3.15.** а) найти произведение элементов вектора, две последние цифры которого не равны 4;  
б) найти сумму положительных элементов вектора, расположенных до максимального элемента.
- 3.16.** а) найти среднее арифметическое элементов вектора, сумма двух последних цифр которых не делится на 4;  
б) найти произведение, оканчивающихся на 11 элементов вектора, расположенных после последнего наименьшего элемента.
- 3.17.** а) найти произведение элементов вектора, две последние цифры которых не 7 и не 8;  
б) найти среднее арифметическое элементов вектора, не оканчивающихся на 15, расположенных до первого наибольшего элемента.
- 3.18.** а) найти сумму элементов вектора, произведение двух последних цифр у которых четно;

- б) найти среднее арифметическое элементов вектора, расположенных после последнего кратного 7 элемента.
- 3.19.** а) найти количество элементов вектора, сумма трех последних цифр у которых не превосходит  $C$ ;  
б) найти сумму целых частей элементов вектора, расположенных до первого отрицательного элемента.
- 3.20.** а) найти количество элементов вектора, три последние цифры которого не более 4;  
б) найти сумму нечетных элементов вектора, расположенных после максимального элемента.

## ЛАБОРАТОРНАЯ РАБОТА №4

### *Использование математических функций*

#### *Содержание*

Запустить среду RStudio. Сформировать вектор  $x$ , а затем сформировать вектор  $y$  в соответствии с некоторой заданной функцией. Например:

Функция  $y = \frac{x^2+25}{x}$  на отрезке  $[1; 10]$ .

```
x<-seq(1,10,by=0.1)
y<-(x*x+25)/x
> y
```

```
[1] 26.00000 23.82727 22.03333 20.53077 19.25714 18.16667
[7] 17.22500 16.40588 15.68889 15.05789 14.50000 14.00476
[13] 13.56364 13.16957 12.81667 12.50000 12.21538 11.95926
[19] 11.72857 11.52069 11.33333 11.16452 11.01250 10.87576
[25] 10.75294 10.64286 10.54444 10.45676 10.37895 10.31026
[31] 10.25000 10.19756 10.15238 10.11395 10.08182 10.05556
[37] 10.03478 10.01915 10.00833 10.00204 10.00000 10.00196
[43] 10.00769 10.01698 10.02963 10.04545 10.06429 10.08596
[49] 10.11034 10.13729 10.16667 10.19836 10.23226 10.26825
[55] 10.30625 10.34615 10.38788 10.43134 10.47647 10.52319
[61] 10.57143 10.62113 10.67222 10.72466 10.77838 10.83333
[67] 10.88947 10.94675 11.00513 11.06456 11.12500 11.18642
[73] 11.24878 11.31205 11.37619 11.44118 11.50698 11.57356
[79] 11.64091 11.70899 11.77778 11.84725 11.91739 11.98817
[85] 12.05957 12.13158 12.20417 12.27732 12.35102 12.42525
[91] 12.50000
```

Затем для вектора значений функции вычислить наибольшее, наименьшее значения, среднее значение и сумму. Использовать функции: `max()`, `min()`, `sum()`, `mean()`.

Выполнить различные варианты округления полученных величин, используя функции `ceiling(x)`, `floor(x)`, `trunc(x)`, `round(x, dig)`, `signif(x, digits = 6)`.



**Пример 1.** Используя команду цикла, найти количество значений функции, равных 10.

```
> x<-seq(1,10,by=0.1)
> y<-(x*x+25)/x
> k=integer(1)
> for (i in 1:100)
+   if (round(y[i])==10)   {k=k+1}
> k
[1] 32
```

Необходимо построить график функции с помощью команды `curve`. Определить точку  $x$ , в которой достигается максимум или минимум функции с помощью цикла. Решить задачу аналитически и сравнить ответ.

Синтаксис функции:

`curve(функция, начальное значение  $x$ , конечное значение  $x$ )`

Например:

`curve (cos(x*x),1,10)`

**Пример 2.**

Найти точку максимума функции  $y = \sqrt{-6 + 12x - x^2}$ .

```
> x<-seq(1,10,by=0.1)
> y<-sqrt(-6+12*x-x*x)
> m<-max(y)
> m
[1] 5.477226
> y
[1]  2.236068  2.447448  2.638181  2.812472  2.973214  3.122499
[7]  3.261901  3.392639  3.515679  3.631804  3.741657  3.845777
[13]  3.944617  4.038564  4.127953  4.213075  4.294182  4.371499
[19]  4.445222  4.515529  4.582576  4.646504  4.707441  4.765501
[25]  4.820788  4.873397  4.923413  4.970915  5.015974  5.058656
[31]  5.099020  5.137120  5.173007  5.206726  5.238320  5.267827
[37]  5.295281  5.320714  5.344156  5.365631  5.385165  5.402777
[43]  5.418487  5.432311  5.444263  5.454356  5.462600  5.469004
[49]  5.473573  5.476313  5.477226  5.476313  5.473573  5.469004
[55]  5.462600  5.454356  5.444263  5.432311  5.418487  5.402777
[61]  5.385165  5.365631  5.344156  5.320714  5.295281  5.267827
```

```

[67] 5.238320 5.206726 5.173007 5.137120 5.099020 5.058656
[73] 5.015974 4.970915 4.923413 4.873397 4.820788 4.765501
[79] 4.707441 4.646504 4.582576 4.515529 4.445222 4.371499
[85] 4.294182 4.213075 4.127953 4.038564 3.944617 3.845777
[91] 3.741657
> k=integer(1)
> curve(sqrt(-6+12*x-x*x),1,10)
> for (i in 1:100)
+   if (y[i]==m) {k=x[i]; break}
> k
[1] 6

```

### ***Построение нескольких графиков***

Рассмотрим пример построения графиков функции на заданном отрезке. В этом случае удобно использовать функцию `par()`.

**Пример 3.** Вывести различные графики функций. Графики функции необходимо отобразить в шести окнах.

Сначала необходимо выделить окна для вывода графиков.

Зададим шесть окон:

```
par(mfrow=c(3,2))
```

Затем укажем вектор  $x$  и функцию для  $y$ :

```

x=seq(-2*pi,2*pi,by=0.01)
y=cos(x)

```

Далее с помощью функции `plot` выведем различные графики:

```

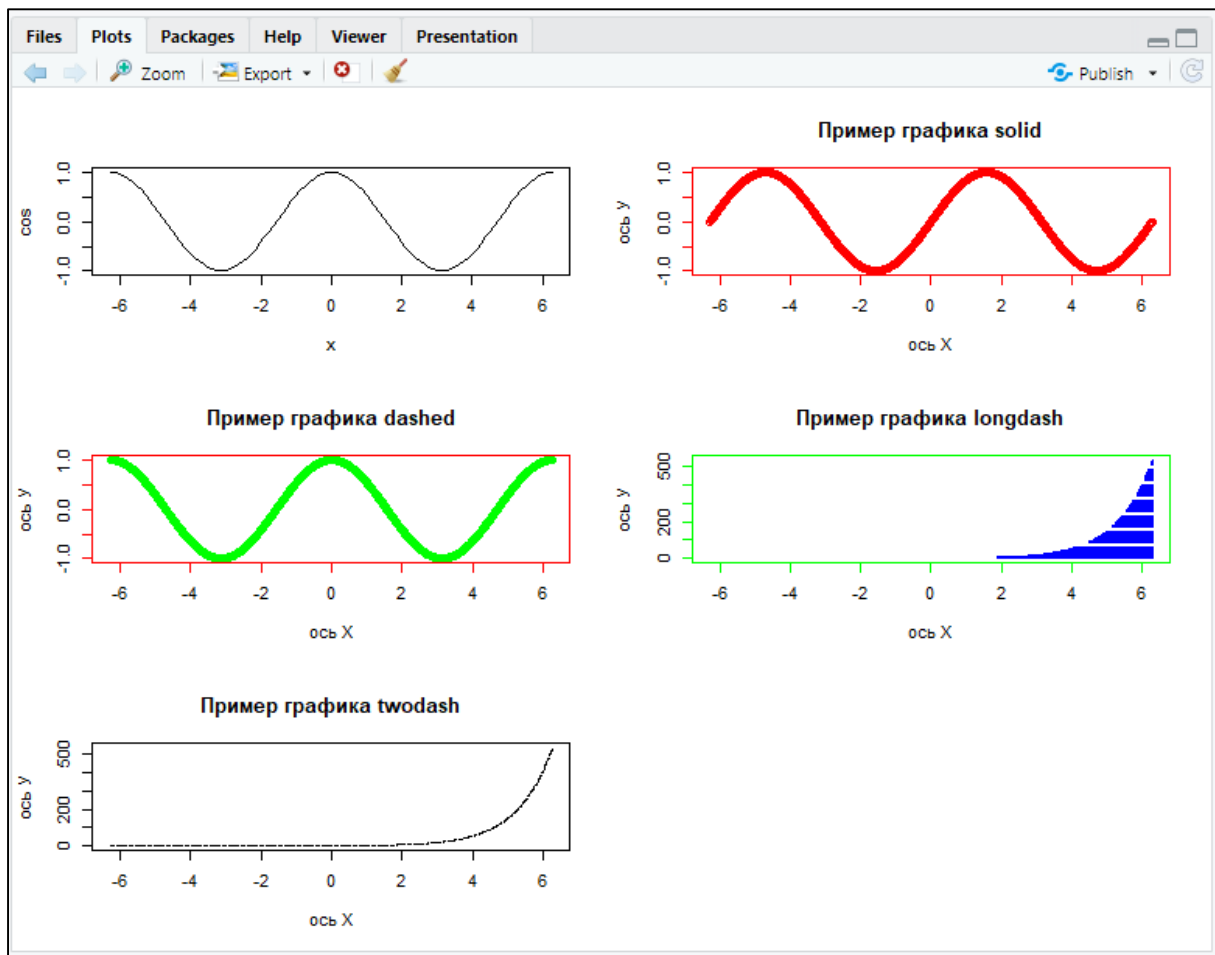
> # зададим шесть окон
> par(mfrow=c(3,2))
> #укажем вектор x и функцию для y:
> x=seq(-2*pi,2*pi,by=0.01)
> y=cos(x)
> #с помощью функции plot выведем различные графики:
> plot(x,y,type="l",lty="dotdash",xlab="ось X",
ylab="ось Y", main="Пример графика dotdash", col="blue",
fg="red")
> plot(cos,-2*pi,2*pi)

```

```

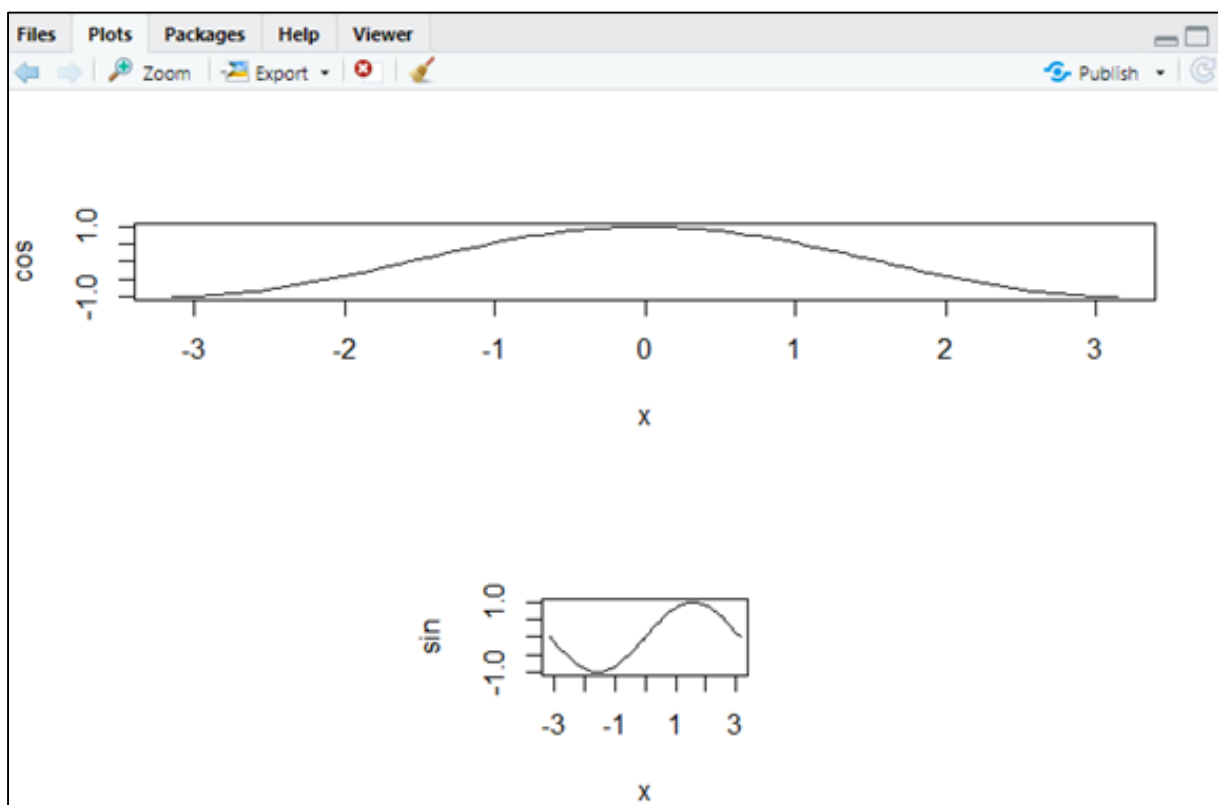
> plot(x,sin(x),type="p",lty="solid",xlab="ось X",
ylab="ось Y", main="Пример графика solid",col="red",
fg="red")
> plot(x,y,type="b",lty="dashed",xlab="ось X",
ylab="ось Y", main="Пример графика dashed",col="green",
fg="red")
> y=exp(x)+cos(x)
> plot(x,y,type="h",lty="longdash",xlab="ось X",
ylab="ось Y", main="Пример графика longdash",col="blue",
fg="green")
> plot(x,y,type="s",lty="twodash",xlab="ось X",
ylab="ось Y", main="Пример графика twodash")

```



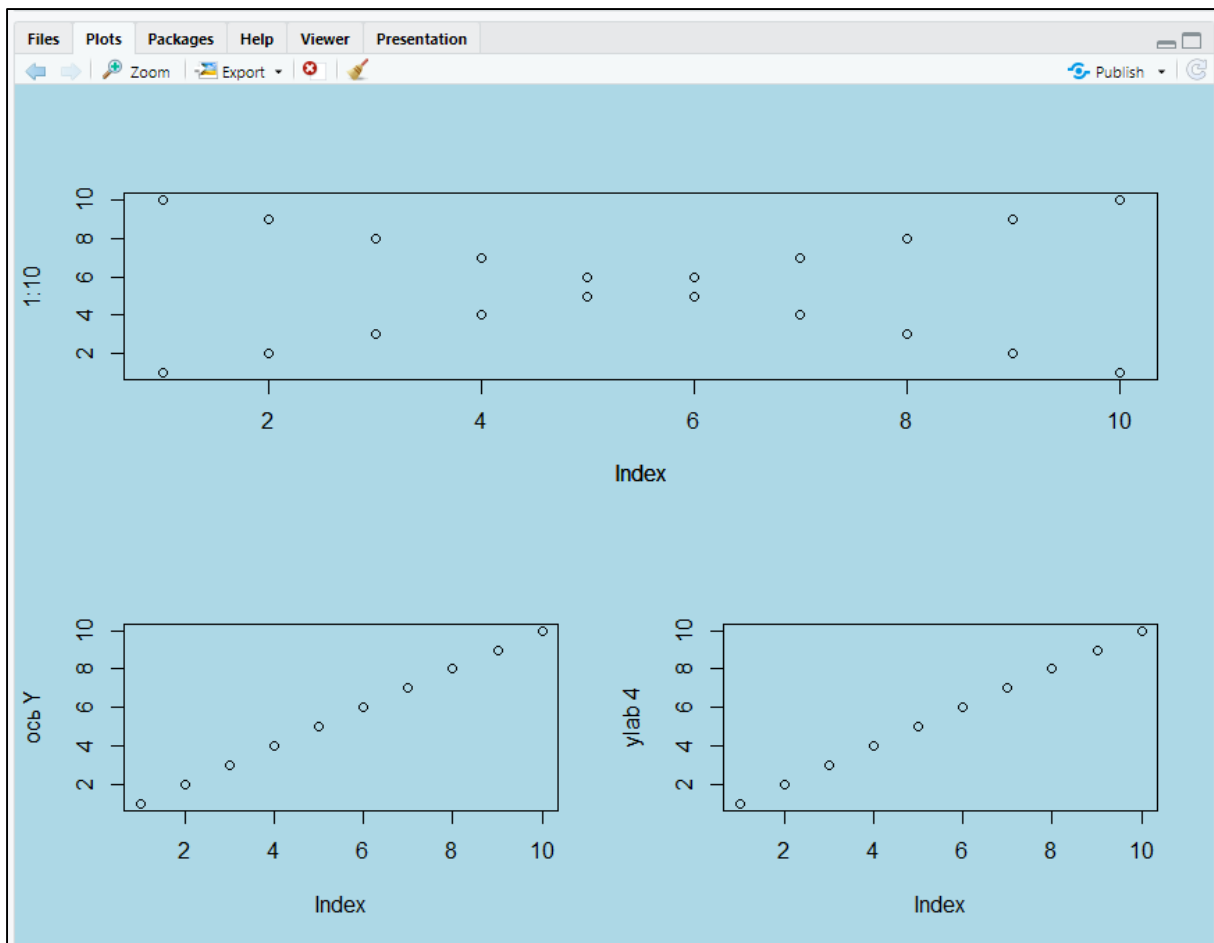
**Пример 4.** Продемонстрируем использование функции `screen()`.

```
> par(bg = "white")
> split.screen(c(2,1))
[1] 1 2
> #окно разбивается на два подокна
> split.screen(c(1,3), screen = 2)
[1] 3 4 5
> #нижнее разбивается еще на три
> screen(1)
> #в первом нижнем окне выводится график
> plot(cos,-pi,pi)
> screen(4)
> plot(sin,-pi,pi)
> #выход из многооконного режима
> close.screen(all = TRUE)
```



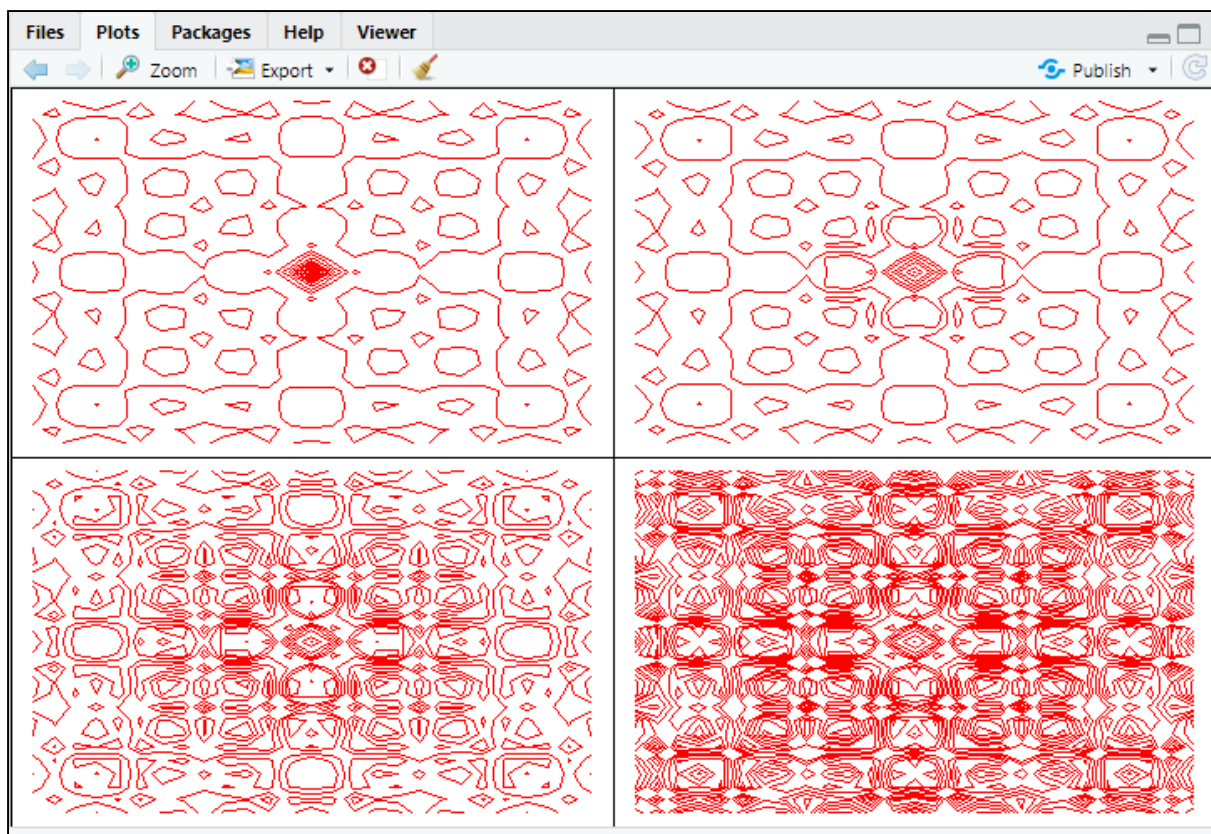
**Пример 5.** Продемонстрируем процесс перемещения между окнами.

```
> par(bg = "lightblue")
> split.screen(c(2,1))
[1] 1 2
> split.screen(c(1,2),2)
[1] 3 4
> # Окно разбивается на два, а затем нижнее еще на три.
> # Активным является первое из трех нижних окон.
> # Построим в нем график
> plot(1:10)
> # Добавим название оси y, для этого стираем
> # созданный график и снова его строим
> erase.screen()
> plot(1:10, ylab="ось Y")
> # Переходим к первому окну, а затем к четвертому
> screen(1)
> plot(1:10)
> screen(4)
> plot(1:10, ylab="ylab 4")
> # Возвращаемся к первому окну, но не стираем его
> screen(1, FALSE)
> plot(10:1, axes=FALSE, lty=2, ylab="")
> # Выход из режима разбиения
> close.screen(all = TRUE)
```



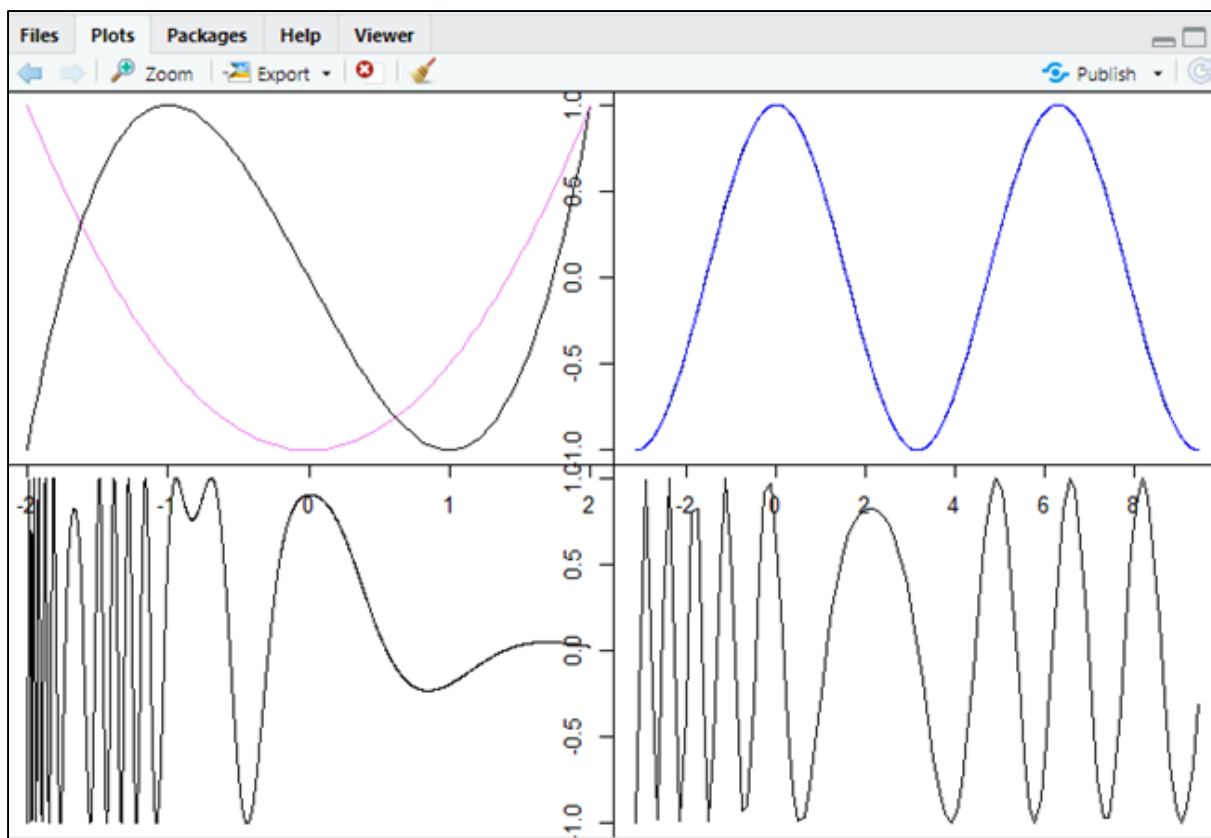
### Пример 6. Демонстрация работы функции `Contour()`.

```
> # Зададим значения векторов x и y
> x <- y <- seq(-4*pi, 4*pi, len = 27)
> # И определим на них функцию
> r <- sqrt(outer(x^2, y^2, "+"))
> # Разобьем графическое окно на 4 части и
> # зададим размеры отступов
> opar <- par(mfrow = c(2, 2), mar = rep(0, 4))
> # Для различных значений параметра f=i*pi, i от 0 до 3
> # построим график некоторой функции z
> for(f in pi^(0:3))
+   contour(cos(r^2)*exp(-r/f), drawlabels = FALSE,
+ axes = FALSE, frame = TRUE)
```



**Пример 7.** Построение различных графиков функций с помощью функции `curve()`.

```
> # Задаем окно и разбиение на 4
> par(mfrow=c(2,2))
> # Рисуем график 1 и в этом же окне график 2
> curve(x^3-3*x, -2, 2)
> curve(x^2-2, add = TRUE, col = "violet")
> # В следующем окне выводим график cos обычный
> plot(cos, -pi, 3*pi)
> # Затем поверх та же функция cos, но с разбиением
> plot(cos, xlim = c(-pi,3*pi), n = 1001, col = "blue",
add=TRUE)
> # Отдельно задаем функцию в переменную f1
> f1 <- function(x) sin(cos(x)*exp(-x/2))
> # Выводим график для f1 в двух окнах разными функциями
> # на разных интервалах
> curve(f1, -8, 7, n=2001)
> plot (f1, -8, -5)
```



### ***Задание***

1. Используя данные из Таблицы исходных данных построить вектор  $y$  на основе некоторой функции, например, вычисляющей 15% от имеющегося столбца (например, стоимость билета). Либо рассмотреть два вектора  $x_1$  (цена за штуку) и  $x_2$  (количество штук в упаковке), вычислить  $y$  по функции, определяющей стоимость продажи.

Для вектора  $y$  найти:

- наибольшее / наименьшее значение;
- среднее значение, округленное до 3 знаков после запятой;
- построить график функции с помощью команды `curve`. Определить точку  $x$ , в которой достигается максимум или минимум функции. Решить задачу аналитически и сравнить ответ.

2. Используя функции `par()` и `screen()` (две отдельные программные реализации), разработать структуру окон (не менее 4) с графиками различных типов (должны отличаться: тип и цвет линии, подписи осей и заголовки и т.д.). В соответствии с индивидуальной тематикой и сгенерированной Таблицей



исходных данных необходимо построить различные графики зависимостей данных.

3. Построить линию зависимости между двумя столбцами данных, используя метод наименьших квадратов (функция `lm`) и функцию `abline` (например, определить зависимость между стоимостью билета и длительностью полета).

4. Используя различные исходные функции с помощью функции `Contour()`, получить шесть окон узора.

5. Выполнить индивидуальное задание.

***Индивидуальное задание***  
***(поиск максимума / минимума;***  
***построение графика функции)***

*Вычислить наибольшее/наименьшее значение функции.  
Построить график функции.*

**4.1.** Найти наибольшее значение и построить график функции  
 $y = \log_5(4 - 2x - x^2) + 3$ .

**4.2.** Найти точку минимума функции  $y = \sqrt{x^2 + 6x + 12}$ .

**4.3.** Найти точку минимума функции  $y = 7^{x^2+2x+3}$ .

**4.4.** Найти наименьшее значение функции  $y = \sqrt{x^2 - 6x + 13}$ .

**4.5.** Найти точку максимума функции  $y = 11^{6x-x^2}$ .

**4.6.** Найти точку минимума функции  $y = \sqrt{x^2 - 6x + 11}$ .

**4.7.** Найти наименьшее значение функции  
 $y = \log_3(x^2 - 6x + 10) + 2$ .

**4.8.** Найти наибольшее значение функции  $y = \sqrt{5 - 4x - x^2}$ .

**4.9.** Найти наименьшее значение функции  $y = 2^{x^2+2x+5}$ .

**4.10.** Найти точку максимума функции  
 $y = \log_2(2 + 2x - x^2) - 2$ .

**4.11.** Найти наибольшее значение функции  
 $y = \log_{\frac{1}{3}}(x^2 + 6x + 12)$  на отрезке  $[-19; -1]$ .

**4.12.** Найти точку минимума функции  
 $y = \log_5(x^2 - 6x + 12) + 2$ .

**4.13.** Найти наибольшее значение функции  $y = 3^{-7-6x-x^2}$ .

**4.14.** Найти наименьшее значение функции  $y = 7^{x^2-2x+3}$ .

- 4.15.** Найти точку максимума функции  $y = \frac{98}{x} + 2x + 15$ .
- 4.16.** Найти наименьшее значение функции  $y = 3x - \ln(x + 3)^3$  на отрезке  $[-2,5; 0]$ .
- 4.17.** Найти наибольшее значение функции  $y = \ln(11x) - 11x + 9$  на отрезке  $\left[\frac{1}{22}; \frac{5}{22}\right]$ .
- 4.18.** Найти точку максимума функции  $y = \frac{11}{x} + 2x^2 + 15$ .
- 4.19.** Найти наименьшее значение функции  $y = 11x - \ln(x + 5)^5$  на отрезке  $[-4,5; 5]$ .
- 4.20.** Найти наибольшее значение функции  $y = \ln(5x + 6) - 7x^3 + 3$  на отрезке  $[0; 5]$ .

## ЛАБОРАТОРНАЯ РАБОТА №5

### *Матрицы*

Рассмотрим создание и обработку матриц.

```
> A=matrix(1:12,nrow=3);A
      [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
> B=matrix(13:24,nrow=3);B
      [,1] [,2] [,3] [,4]
[1,]   13   16   19   22
[2,]   14   17   20   23
[3,]   15   18   21   24
> Z=cbind(A,B);Z
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,]    1    4    7   10   13   16   19   22
[2,]    2    5    8   11   14   17   20   23
[3,]    3    6    9   12   15   18   21   24
```

Последняя функция приписывает матрицу справа к исходной. Аналогично работает функция `rbind()`, приписывая матрицу снизу к исходной.

```
> Z=rbind(A,B);Z
      [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
[4,]   13   16   19   22
[5,]   14   17   20   23
[6,]   15   18   21   24
```

Для задания диагональной матрицы используют функцию `diag(x, nrow, ncol)`.

```
> diag(1,3,3)
      [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
```

Также можно задать единичную матрицу.

```
> diag(nrow=4)
      [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]    0    1    0    0
[3,]    0    0    1    0
[4,]    0    0    0    1
```

Для редактирования матрицы можно использовать функцию `fix(A)`.

```
A=matrix(1:16,nrow=4);A
      [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16
> fix(A)
```

	col1	col2	col3	col4	var5	var6	var7	var8
1	1	5	9	13				
2	2	6	10	14				
3	3	7	11	15				
4	4	8	12	16				
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								

При выполнении арифметических операций над матрицами необходимо соблюдать размерность аргументов (количество строк и столбцов).

```
> A = matrix(1:9, nrow = 3);A
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
```

```

[3,]      3      6      9
> B = matrix(-(1:9), ncol = 3,byrow=T);B
      [,1] [,2] [,3]
[1,]    -1    -2    -3
[2,]    -4    -5    -6
[3,]    -7    -8    -9
> A + B
      [,1] [,2] [,3]
[1,]      0      2      4
[2,]     -2      0      2
[3,]     -4     -2      0
> A+3
      [,1] [,2] [,3]
[1,]      4      7     10
[2,]      5      8     11
[3,]      6      9     12
> B+3
      [,1] [,2] [,3]
[1,]      2      1      0
[2,]     -1     -2     -3
[3,]     -4     -5     -6
> (1:3)*A
      [,1] [,2] [,3]
[1,]      1      4      7
[2,]      4     10     16
[3,]      9     18     27
> A*(1:3)
      [,1] [,2] [,3]
[1,]      1      4      7
[2,]      4     10     16
[3,]      9     18     27
> B^2
      [,1] [,2] [,3]
[1,]      1      4      9
[2,]     16     25     36
[3,]     49     64     81

```

Применение математических функций применяется к матрице поэлементно.

```

> sqrt(A)
      [,1]      [,2]      [,3]
[1,] 1.000000 2.000000 2.645751
[2,] 1.414214 2.236068 2.828427
[3,] 1.732051 2.449490 3.000000
> log(abs(B))
      [,1]      [,2]      [,3]
[1,] 0.000000 0.6931472 1.098612
[2,] 1.386294 1.6094379 1.791759
[3,] 1.945910 2.0794415 2.197225

```

Функция `outer(x, y, "операция")` применяет заданную операцию к каждой паре элементов векторов `x` и `y`. Получим матрицу, число строк матрицы соответствует длине вектора `x`, число столбцов – длине вектора `y`.

```

> x =1:5; x
[1] 1 2 3 4 5
> y =-2:3;y
[1] -2 -1 0 1 2 3
> outer(x, y, "*")
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,] -2 -1 0 1 2 3
[2,] -4 -2 0 2 4 6
[3,] -6 -3 0 3 6 9
[4,] -8 -4 0 4 8 12
[5,] -10 -5 0 5 10 15
> outer(x, y, "^")
      [,1]      [,2] [,3] [,4] [,5] [,6]
[1,] 1.0000000 1.0000000 1 1 1 1
[2,] 0.2500000 0.5000000 1 2 4 8
[3,] 0.1111111 0.3333333 1 3 9 27
[4,] 0.0625000 0.2500000 1 4 16 64
[5,] 0.0400000 0.2000000 1 5 25 125

```

Транспонирование матрицы выполняется функцией `t(A)`, а матричное произведение – операцией `%*%`.

```

> A = matrix(1:9, nrow = 3);
> B = matrix(-(1:9), ncol = 3, byrow=T)
> A%*%B

```

```

      [,1] [,2] [,3]
[1,]  -66  -78  -90
[2,]  -78  -93 -108
[3,]  -90 -108 -126
> B%*%A
      [,1] [,2] [,3]
[1,]  -14  -32  -50
[2,]  -32  -77 -122
[3,]  -50 -122 -194

```

Для решения системы линейных уравнений  $Ax=B$  с квадратной невырожденной матрицей  $A$  используется функция `solve(A, B)`.

```

> A = matrix(c(3,4,4,4), nrow = 2);A
      [,1] [,2]
[1,]     3     4
[2,]     4     4
> b=c(1,0)
> solve(A,b)
[1] -1  1

```

Функция `det(A)` находит определитель матрицы  $A$ , `solve(A)` – обратную матрицу.

```

> det(A)
[1] -4
> solve(A)
      [,1] [,2]
[1,]    -1  1.00
[2,]     1 -0.75

```

Дополнительно можно использовать функции:

Имя функции	Результат
<code>colSums(X, na.rm)</code>	Сумма элементов по столбцам
<code>rowSums(X, na.rm)</code>	Сумма элементов по строкам
<code>colMeans(X na.rm)</code>	Среднее значение по столбцам
<code>rowMeans(X na.rm)</code>	Среднее значение по строкам

Здесь  $X$  – матрица, `na.rm` – логический аргумент, определяющий, нужно ли убирать из рассмотрения `na`, по умолчанию используется значение `FALSE`.

```
> A=matrix(1:12,nrow=3);A
      [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
> colSums(A)
[1]  6 15 24 33
> rowSums(A)
[1] 22 26 30
> colMeans(A)
[1]  2  5  8 11
> rowMeans(A)
[1] 5.5 6.5 7.5
```

### Задания

Решите систему уравнений:

$$1. \begin{cases} x + 3y = 7 \\ 2x + 4y = 10 \end{cases} \quad 2. \begin{cases} x - 2y = 5 \\ 3x + y = 8 \end{cases} \quad 3. \begin{cases} x + 3y + 2z = 5 \\ 2x + 4y - 2z = 8 \\ 3x - 2y - z = 0 \end{cases}$$

Вычислить определитель

$$4. \begin{vmatrix} 3 & -2 \\ 4 & 6 \end{vmatrix} \quad 5. \begin{vmatrix} 2 & 3 \\ 6 & -10 \end{vmatrix} \quad 6. \begin{vmatrix} 3 & -2 \\ -4 & 5 \end{vmatrix} \quad 7. \begin{vmatrix} 2 & -2 \\ 1 & -1 \end{vmatrix}$$

8. Даны матрицы

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & -2 & -4 & 5 \end{bmatrix} \quad B = \begin{bmatrix} 1 & -1 & 1 & 2 \\ 2 & -2 & -4 & 5 \end{bmatrix}.$$

Найти:  $C = 3A - 4B$ . В ответе запишите наименьший элемент матрицы  $C$ .

9. Даны матрицы

$$A = \begin{bmatrix} 2 & -3 \\ 4 & 5 \end{bmatrix} \quad B = \begin{bmatrix} 1 & -3 & -2 \\ 2 & 4 & -1 \end{bmatrix}.$$

Найти матрицу  $C = A \cdot B$ .



10. Найти куб матрицы  $A = \begin{bmatrix} 1 & -1 \\ 3 & 2 \end{bmatrix}$ .

11. Даны матрицы

$$A = \begin{bmatrix} 2 & -1 & 0 \\ 4 & 3 & 1 \end{bmatrix} \quad B = \begin{bmatrix} -3 & 2 & 4 \\ 5 & -6 & 3 \end{bmatrix}.$$

Найти:  $C = 2A - 3B$ . В ответе запишите сумму элементов матрицы  $C$ .

12. Выполнить индивидуальное задание.

### ***Индивидуальное задание (матрицы)***

Соответствие индивидуальных номеров и заданий указано в таблице:

Вариант	Задания			
1	5.1	5.21	5.41	5.61
2	5.2	5.22	5.42	5.62
3	5.3	5.23	5.43	5.63
4	5.4	5.24	5.44	5.64
5	5.5	5.25	5.45	5.65
6	5.6	5.26	5.46	5.66
7	5.7	5.27	5.47	5.67
8	5.8	5.28	5.48	5.68
9	5.9	5.29	5.49	5.69
10	5.10	5.30	5.50	5.70
11	5.11	5.31	5.51	5.71
12	5.12	5.32	5.52	5.72
13	5.13	5.33	5.53	5.73
14	5.14	5.34	5.54	5.74
15	5.15	5.35	5.55	5.75
16	5.16	5.36	5.56	5.76
17	5.17	5.37	5.57	5.77
18	5.18	5.38	5.58	5.78
19	5.19	5.39	5.59	5.79
20	5.20	5.40	5.60	5.80

Вычислить определитель

$$5.1. \begin{vmatrix} -1 & -2 \\ 5 & -4 \end{vmatrix} \quad 5.2. \begin{vmatrix} 3 & -7 \\ 1 & -4 \end{vmatrix} \quad 5.3. \begin{vmatrix} 1 & 4 \\ -2 & 8 \end{vmatrix} \quad 5.4. \begin{vmatrix} a & -1 \\ a^2 & a \end{vmatrix}$$

$$5.5. \begin{vmatrix} 2 & 3 & 4 \\ 5 & -2 & 1 \\ 1 & 2 & 3 \end{vmatrix} \quad 5.6. \begin{vmatrix} -2 & 2 & -1 \\ 4 & 2 & 4 \\ 1 & 3 & -2 \end{vmatrix} \quad 5.7. \begin{vmatrix} 2 & 3 & 2 \\ 0 & 0 & 0 \\ 6 & 7 & 8 \end{vmatrix}$$

$$5.8. \begin{vmatrix} -3 & 2 & 4 \\ 4 & 2 & 4 \\ 4 & 6 & 12 \end{vmatrix} \quad 5.9. \begin{vmatrix} 1 & 2 & 5 \\ 3 & -4 & 7 \\ -3 & 12 & -15 \end{vmatrix} \quad 5.10. \begin{vmatrix} 12 & 6 & -4 \\ 6 & 4 & 4 \\ 3 & 2 & 8 \end{vmatrix}$$

$$5.11. \begin{vmatrix} 2 & -3 & 1 \\ 6 & -6 & 2 \\ 2 & -1 & 2 \end{vmatrix} \quad 5.12. \begin{vmatrix} 4 & 6 & -2 & 4 \\ 1 & 2 & -3 & 1 \\ 4 & -2 & 1 & 0 \\ 6 & 4 & 4 & 6 \end{vmatrix}$$

$$5.13. \begin{vmatrix} 4 & 8 & -4 & 3 \\ 8 & 4 & -2 & 1 \\ -3 & -4 & 2 & 3 \\ 3 & 7 & -3 & 3 \end{vmatrix} \quad 5.14. \begin{vmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 4 & 1 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{vmatrix}$$

$$5.15. \text{Даны матрицы: } P = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 2 & 3 & 1 \end{bmatrix} \text{ и } Q = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Найти сумму P+Q.

$$5.16. Q = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad R = \begin{bmatrix} 1 & 1 & 3 \\ 2 & 1 & 2 \\ 1 & 3 & 1 \end{bmatrix}$$

Найти  $5*Q-4*R$

5.17. Вычислить определитель матрицы:

$$\begin{vmatrix} 13 & 13 & 10 \\ 10 & 13 & 13 \\ 13 & 10 & 13 \end{vmatrix}$$

$$5.18. \text{Даны матрицы: } A = \begin{bmatrix} 0 & 1 \\ 3 & 2 \end{bmatrix} \text{ и } B = \begin{bmatrix} -2 & 0 \\ 2 & 3 \end{bmatrix}$$

Найти: а)  $5A - B$ ; б)  $3A' - 2B$ ; в)  $AB$

Найти произведение матриц

$$5.19. \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 0 \\ 1 & 0 & 3 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ 2 \end{bmatrix} \quad 5.20. \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & -2 \\ 2 & 1 & -1 \end{bmatrix}$$

5.21. Вычислить матрицу  $D = (AB)' - C^2$ , где

$$A = \begin{bmatrix} 4 & 3 \\ 7 & 5 \end{bmatrix}, B = \begin{bmatrix} -28 & 93 \\ 38 & -126 \end{bmatrix}, C = \begin{bmatrix} 1 & 3 \\ 0 & 4 \end{bmatrix}.$$

5.22. Даны матрицы  $A = \begin{bmatrix} 1 & 3 & 1 \\ 2 & 0 & 4 \\ 1 & 2 & 3 \end{bmatrix}$   $B = \begin{bmatrix} 2 & 1 & 0 \\ 1 & -1 & 2 \\ 3 & 2 & 1 \end{bmatrix}$ . Найти:

а)  $A^2 + 2B - 3E$ ; б)  $AB - BA$ ; в)  $(A - 2B)^2$ ;  
г)  $A^2 - B^2$ ; д)  $(A - B)(A + B)$

5.23. Найти произведение матриц  $ABC - 3E$ , где

$$A = \begin{bmatrix} 1 & 2 & -3 \\ 1 & 0 & 2 \\ 4 & 5 & 3 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}, C = [2 \quad 0 \quad 5], E - \text{единичная}.$$

5.24. Вычислить  $A^2$ , если  $A = \begin{bmatrix} 1 & 1 & -1 \\ 3 & -1 & 2 \\ 2 & -1 & 0 \end{bmatrix}$ .

5.25. Найти произведение матриц

$$A = \begin{bmatrix} 1 & -3 & 5 \\ 6 & -5 & 9 \\ 5 & 2 & 2 \end{bmatrix} \text{ и } B = \begin{bmatrix} 4 & -5 & 1 \\ 2 & 7 & -2 \\ 5 & 2 & 2 \end{bmatrix}.$$

Найти обратные матрицы для следующих матриц:

$$5.26. \begin{bmatrix} 1 & -3 \\ 3 & -4 \end{bmatrix} \quad 5.27. \begin{bmatrix} 2 & -1 \\ 1 & -4 \end{bmatrix} \quad 5.28. \begin{bmatrix} 5 & 1 \\ 7 & -3 \end{bmatrix} \quad 5.29. \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$5.30. \begin{bmatrix} 3 & 2 & 2 \\ 1 & 3 & 1 \\ 5 & 3 & 4 \end{bmatrix} \quad 5.31. \begin{bmatrix} 3 & 2 & 2 \\ 1 & -2 & 3 \\ 0 & 1 & -1 \end{bmatrix} \quad 5.32. \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix}$$

$$5.33. \begin{bmatrix} 1 & 2 & 3 \\ -3 & -2 & -4 \\ 2 & -1 & 0 \end{bmatrix} \quad 5.34. \begin{bmatrix} 14 & 15 & 13 \\ 13 & 14 & 15 \\ 15 & 13 & 14 \end{bmatrix} \quad 5.35. \begin{bmatrix} 0 & 1 & -2 \\ -1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix}.$$

**5.36.** Вычислить определитель матрицы

$$A = \begin{bmatrix} 1 & 0 & -1 & -1 \\ 0 & -1 & -1 & 1 \\ 1 & 2 & 3 & 4 \\ -1 & -1 & 1 & 0 \end{bmatrix}.$$

**5.37.** Даны матрицы

$$A = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 3 & 1 \\ 2 & 1 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 & 4 \\ 5 & 1 & 2 \\ 3 & -1 & 2 \end{bmatrix}, \quad C = \begin{bmatrix} 2 \\ 2 \\ -1 \end{bmatrix}.$$

Найти: а)  $D = -AC + 2B^T C$ ; б)  $G = C^T 2A^T - 2C^T B$ .

Вычислить определители:

$$5.38. D_1 = \begin{bmatrix} 2 & -1 & 3 & 0 \\ 4 & 1 & 2 & -1 \\ -3 & 0 & 4 & 1 \\ 1 & 1 & 0 & 3 \end{bmatrix}. \quad 5.39. D_2 = \begin{bmatrix} -2 & 0 & -1 & -1 \\ 3 & 1 & -2 & 1 \\ 0 & 4 & -1 & 3 \\ 1 & 1 & -2 & 1 \end{bmatrix}.$$

$$5.40. D_1 = \begin{bmatrix} 2/3 & 3/8 & -3 & -4 \\ 2/3 & 1/8 & -1 & -2 \\ 2 & 1/4 & 1 & 0 \\ 2/3 & 3/8 & 0 & -5 \end{bmatrix}.$$

$$5.41. D_2 = \begin{bmatrix} 8/3 & 7/5 & 2/5 & 0 \\ -8/3 & 2/5 & 7/5 & 10 \\ 4/3 & 4/5 & 4/5 & 5 \\ 0 & 4/5 & -3/5 & 2 \end{bmatrix}.$$

Докажите, что данная матрица имеет обратную и найдите её.  
Выполните проверку:

5.42.  $\begin{bmatrix} 7 & -8 \\ 5 & -3 \end{bmatrix}$

5.43.  $\begin{bmatrix} -9 & 6 \\ 0 & 2 \end{bmatrix}$

5.44. Дана матрица  $I = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 3 & 1 \\ 2 & 1 & 3 \end{bmatrix}$ . Найдите элементы обратной матрицы:  $b_{13}$ ,  $b_{12}$ ,  $b_{31}$ .

5.45. Докажите, что матрица  $A = \begin{bmatrix} 1 & 3 & -5 & -1 \\ 0 & -2 & 3 & 6 \\ 5 & -4 & 2 & 0 \\ -7 & 1 & 5 & 1 \end{bmatrix}$  имеет обратную. Найдите элементы обратной матрицы:  $b_{12}$ ,  $b_{23}$ ,  $b_{34}$ ,  $b_{43}$ .

5.46. Докажите, что матрица  $A = \begin{bmatrix} 3 & -1 & 2 \\ -2 & 1 & 1 \\ 1 & -2 & -3 \end{bmatrix}$  имеет обратную  $A^{-1}$  и найдите её. Выполните проверку.

5.47. Докажите, что матрица  $A = \begin{bmatrix} -1 & 8 & 4 \\ 3 & -1 & 2 \\ 0 & -5 & -7 \end{bmatrix}$  имеет обратную  $A^{-1}$  и найдите её. Выполните проверку.

5.48. Решите матричные уравнения  $AX_1 = B$  и  $X_2A = B$  если  $A = \begin{bmatrix} 1 & 3 \\ 2 & 5 \end{bmatrix}$  и  $B = \begin{bmatrix} 4 & -1 \\ 3 & 2 \end{bmatrix}$ .

Решите матричные уравнения

5.49.  $\begin{bmatrix} 2 & 7 & 3 \\ 3 & 9 & 4 \\ 1 & 5 & 3 \end{bmatrix} \cdot X = 3 \begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 3 & 4 \end{bmatrix};$

$$5.50. X \cdot \begin{bmatrix} 2 & 7 & 3 \\ 3 & 9 & 4 \\ 1 & 5 & 3 \end{bmatrix} = 3 \begin{bmatrix} 1 & 2 & 1 \\ 3 & 1 & 4 \end{bmatrix};$$

$$5.51. \begin{bmatrix} -1 & 2 & 4 \\ -3 & 1 & 2 \\ -3 & 0 & 1 \end{bmatrix} \cdot X \cdot \begin{bmatrix} 3 & -1 \\ 2 & 1 \end{bmatrix} = - \begin{bmatrix} 7 & 21 \\ 11 & 8 \\ 8 & 4 \end{bmatrix};$$

$$5.52. \begin{bmatrix} 3 & -1 \\ 2 & 1 \end{bmatrix} \cdot X^T \cdot \begin{bmatrix} -1 & 2 & -4 \\ -3 & 1 & 2 \\ 3 & 0 & -1 \end{bmatrix} = \begin{bmatrix} -3 & 11 & -7 \\ 13 & 9 & -8 \end{bmatrix}.$$

Найдите решения следующих систем линейных уравнений, записанных в матричной форме:

$$5.53. \begin{bmatrix} 2 & 3 \\ 4 & -5 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 7 \\ 3 \end{bmatrix}; \quad \text{б) } \begin{bmatrix} 1 & -4 \\ 2 & -3 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 7 \\ 4 \end{bmatrix};$$

$$5.54. \begin{bmatrix} 1 & 2 & 3 \\ 3 & -1 & 4 \\ 1 & 3 & 3 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 \\ 13 \\ 1 \end{bmatrix}; \quad \text{г) } \begin{bmatrix} 2 & -1 & 3 \\ 1 & -2 & 1 \\ 3 & -1 & 3 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}.$$

$$5.55. \text{Решите систему уравнений: } \begin{cases} 2x - 3y = 13 \\ 3x + 5y = -9 \end{cases}.$$

Даны матрицы  $A = \begin{bmatrix} 1 & -2 \\ 3 & -4 \end{bmatrix}$ ,  $B = \begin{bmatrix} 5 & 0 \\ 1 & -2 \end{bmatrix}$ . Найдите:

$$5.56. A + B; \quad 5.57. 2A; \quad 5.58. 2A + 3B; \quad 5.59. 2B - A;$$

$$5.60. AB; \quad 5.61. A^2 + 3E; \quad 5.62. AB - BA.$$

5.63. Дано произведение матриц:

$$\begin{bmatrix} 1 & -2 \\ 4 & 3 \end{bmatrix} \cdot \begin{bmatrix} 2 & 4 & 5 \\ 3 & -1 & 6 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix}$$

Найдите значения  $x_2, x_3, y_1$ .

$$5.64. \text{Решите систему уравнений: } \begin{cases} 2x + 3y + 2z = 1 \\ 3x + 2y + 3z = -1 \\ 2x + 7y + 4z = -2 \end{cases}$$

Даны матрицы

$$A = \begin{bmatrix} 2 & -3 \\ 4 & 5 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 \\ -3 & -4 \end{bmatrix}, \quad C = \begin{bmatrix} 4 & -1 \\ 2 & 1 \end{bmatrix}.$$

Найдите:

**5.65.**  $D = AC + 3BC$ ; **5.66.**  $G = AC + 3CB$ ; **5.67.**  $Q = CA + 3CB$ .

В ответ запишите наибольшие элементы матриц  $D$ ,  $G$  и  $Q$ .

**5.68.** Дано произведение матриц

$$C = \begin{bmatrix} 4 & 2 & -3 & 3 \\ 6 & 4 & -3 & 5 \\ 9 & 2 & -3 & 4 \\ 7 & 6 & -4 & 7 \end{bmatrix} \cdot \begin{bmatrix} 2 & 2 & 2 \\ -1 & -5 & 3 \\ 16 & 24 & 8 \\ 8 & 16 & 0 \end{bmatrix}.$$

Найдите элементы матрицы  $C$ :  $c_{42}$ ,  $c_{13}$ ,  $c_{31}$ .

**5.69.** Решить уравнение: 
$$\begin{cases} x - 2y + 4z = 11 \\ 3x + 2y + 3z = 8. \\ 2x + 5z = 11 \end{cases}$$

**5.70.** Даны матрицы

$$A = \begin{bmatrix} 2 & -4 \\ 3 & 5 \\ -1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 & 7 \\ -3 & -4 & 0 \\ 5 & 2 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 6 & -3 & 9 \\ 4 & -5 & 2 \\ 8 & 1 & 5 \end{bmatrix}.$$

Найти матрицу  $D = A^T C - 2A^T B^T$ .

В ответ запишите наименьший элемент матрицы  $D$ .

**5.71.** Дано произведение матриц

$$C = \begin{bmatrix} 5 & 2 & -3 & 3 \\ 6 & 4 & -3 & 5 \\ 9 & 2 & -3 & 4 \\ 7 & 6 & -4 & 7 \end{bmatrix} \cdot \begin{bmatrix} 2 & 2 & 2 \\ -1 & -5 & 3 \\ 2 & 3 & 1 \\ 1 & 2 & 0 \end{bmatrix}.$$

Найти элементы матрицы  $C$ :  $c_{12}$ ,  $c_{23}$ ,  $c_{31}$ ,  $c_{42}$ ,  $c_{43}$ .

Найдите:

**5.72.**  $D = AB + 2BC$ ; **5.73.**  $G = BC + 3AB$ ; **5.74.**  $Q = CB + 3CA$ .

В ответ запишите наименьшие элементы матриц  $D$ ,  $G$  и  $Q$ .

**5.75.** Даны матрицы

$$A = \begin{bmatrix} 2 & 1 \\ 3 & -5 \\ -1 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 1 & 0 \\ 5 & 2 & 1 \end{bmatrix}, C = \begin{bmatrix} 2 & 1 & -2 \\ 4 & 3 & 2 \\ 3 & 1 & 5 \end{bmatrix}.$$

Найдите матрицу  $D = A^T C - 2A^T B^T$ .

В ответ запишите наибольший элемент матрицы  $D$ .

**5.76.** Дано произведение матриц

$$C = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ 4 & 7 & 8 & 9 \end{bmatrix} \cdot \begin{bmatrix} 2 & 2 & 2 \\ 1 & 1 & 1 \\ 3 & 3 & 3 \\ 1 & 2 & 3 \end{bmatrix}.$$

Найдите элементы матрицы  $C$ :  $c_{11}$ ,  $c_{22}$ ,  $c_{33}$ ,  $c_{44}$ .

Даны матрицы  $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ ,  $B = \begin{bmatrix} -5 & 3 \\ 2 & 1 \end{bmatrix}$ . Найдите:

**5.77.**  $A^2 + 3E$ ; **5.78.**  $2A + 3B$ ; **5.79.**  $B^2 - 2E$ ; **5.80.**  $2B - A$ ;



## ЛАБОРАТОРНАЯ РАБОТА №6

### Функции

Полная форма функции содержит оператор возвращения значения – результата функции.

```
> имя = function(аргументы) {выражения  
return(значение)  
}
```

Полная форма

```
> ff1=function(x){  
+   y=sum(x); z=cumprod(x)  
+   return(y)}  
> ff1(1:10)  
[1] 55
```

Краткая форма

```
> ff=function(x){  
+   y=sum(x)  
+   z=cumprod(x)  
+ }  
> y=ff(1:10); y  
[1] 1 2 6 24 120 720 5040 40320 362880 3628800
```

Продemonстрируем использование функций `apply()`.

**Пример 1.** Создать матрицу X и вычислить суммы значений по строкам и столбцам.

```
> X=matrix(1:25,nrow=5)  
> # Найдем сумму элементов по строкам  
> apply(X,1,sum)  
[1] 55 60 65 70 75  
> # И по столбцам  
> apply(X,2,sum)  
[1] 15 40 65 90 115  
> # В обоих случаях получаем вектора.
```

Далее вычислить корень значений (вывод результата осуществляется по строкам).

```
> # Применим функцию ко всем элементам матрицы.
```

```
> apply(X,1,sqrt)
      [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 1.000000 1.414214 1.732051 2.000000 2.236068
[2,] 2.449490 2.645751 2.828427 3.000000 3.162278
[3,] 3.316625 3.464102 3.605551 3.741657 3.872983
[4,] 4.000000 4.123106 4.242641 4.358899 4.472136
[5,] 4.582576 4.690416 4.795832 4.898979 5.000000
```

Создадим собственную анонимную функцию:

```
> apply(X,1,function(x) x^2-x)
      [,1] [,2] [,3] [,4] [,5]
[1,]    0    2    6   12   20
[2,]   30   42   56   72   90
[3,]  110  132  156  182  210
[4,]  240  272  306  342  380
[5,]  420  462  506  552  600
```

**Пример 2.** Создать список, элементами которого являются три вектора: два числовых и один логический.

```
> x <- list(a = 1:10, beta = exp(-3:3), logic = c(TRUE,FALSE,FALSE,TRUE))
> x
$a
 [1]  1  2  3  4  5  6  7  8  9 10

$beta
 [1] 0.04978707 0.13533528 0.36787944 1.00000000
 [5] 2.71828183 7.38905610 20.08553692

$logic
 [1] TRUE FALSE FALSE TRUE
> sapply(x, sum)
      a      beta      logic
55.00000 31.74588  2.00000
```

**Пример 3.** Для двух целочисленных векторов найти количество четных значений. Использовать собственную функцию.

```
> sum1=function(x)
+ {
+   k=0
+   for (i in 1:10)
+     if (x[i]%%2==0) k=k+1
+   print (k)
+ }
> sum1(2:11)
[1] 5
> x <- list(a = 1:10, beta = 5:15)
> sapply(x,sum1)
[1] 5
[1] 5
   a beta
   5    5
```

### ***Задание***

1. На основе сгенерированных векторов целых чисел FV и PV (не менее 30 значений) вычислить по формулам следующие значения (все формулы *оформить в виде функций*).

### **Сводка основных формул**

Часто используемые в формулах обозначения:

$r(d)$  – годовая процентная (учётная) ставка (в десятичных дробях);  
 $r^{(m)}$  ( $d^{(m)}$ ) – номинальная годовая процентная (учётная) ставка (в десятичных дробях), индекс  $m$  указывает сколько раз в течение года происходит наращение или дисконтирование;

$n, l$  – продолжительность финансовой операции в годах;

$t$  – продолжительность финансовой операции в днях;

$T$  – количество дней в году;

$P$  – первоначальный капитал;

$F$  – наращенный капитал;

$F_n$  – наращенный капитал за  $n$  лет.

Процентная ставка:  $r_t = \frac{FV - PV}{PV}$ ,

где  $PV$  – предоставляемая в долг сумма,

$FV$  – возвращаемая сумма.

Учётная ставка:  $d_t = \frac{FV - PV}{FV}$ .

Соотношение между ставками  $r_t = \frac{d_t}{1 - d_t}$  или  $d_t = \frac{r_t}{1 + r_t}$ .

Дисконт-фактор  $v_t = \frac{PV}{FV}$ .

Индекс роста капитала  $B_t = \frac{FV}{PV}$ .

Построить графики процентной ставки, учетной ставки, соотношения между ставками, дисконт-фактора и индекса роста капитала.

2. Даны действительные числа  $s, t$ . Получить  $g(1.2, s) + g(t, s) - g(2s-1, st)$ ,

где  $g(a, b) = \frac{a^2 + b^2 - 4ab}{a^2 + 5ab + 3b^2 + 4a - b}$ .

Вычисление  $g(a, b)$  оформить как функцию.

3. В соответствии с индивидуальной тематикой и сгенерированной Таблицей исходных данных для некоторого вектора написать функции:

- поиск наибольшего четного значения;
- поиск наименьшего значения, делящегося на 3;
- вычисление произведения порядковых номеров значений, не делящихся на 5 или 8;
- вычисление суммы значений, порядковые номера которых не делятся на 3.

4. Создать матрицу и с помощью функции `apply()` для строк и столбцов вычислить: наибольшее, наименьшее, среднее значение, медиану, произведение. Описать и применить собственную функцию, использующую различные математические функции.

5. С использованием функции `sapply()` для трех числовых векторов (целочисленный, два вектора, заданные функцией) вычислить наибольшее / наименьшее значение, среднее значение, медиану, произведение. Используя собственные функции, найти:

- произведение нечетных элементов векторов;

- сумму последних цифр элементов векторов;
- количество элементов векторов, у которых две последние цифры равны.

6. Выполнить индивидуальное задание.

### ***Индивидуальное задание (функции)***

Для сгенерированного вектора описать и применить функцию. Для результирующего вектора построить график (график должен иметь заголовок, легенду, подписи осей). Содержимое функции соответствует индивидуальному номеру.

- 6.1.** Клиент поместил в банк вклад в сумме  $X$  тыс. р. под 18% годовых с ежеквартальной выплатой простых процентов.
- 6.2.** Клиент поместил в банк вклад в сумме  $X$  тыс. р. под 18% годовых с ежеквартальной выплатой простых процентов. Какую сумму клиент будет получать каждый квартал? Описать функцию вычисления ежеквартальной суммы, применить ее для вектора из 30 сгенерированных значений  $X$ . Построить столбиковую диаграмму.
- 6.3.** Клиент поместил в банк вклад  $X$  тыс. р. под простую процентную ставку 20% годовых. Какая сумма будет на счете клиента через 1–12 месяцев? Описать функцию вычисления суммы, применить ее для вектора из 30 сгенерированных значений  $X$ . Построить столбиковую диаграмму.
- 6.4.** Банк принимает депозиты на 3 месяца по процентной ставке  $X\%$  годовых, на 6 месяцев – по  $Y\%$  годовых и на год – по  $Z\%$  годовых. Определить сумму, которую получит владелец депозита в размере 20 тыс. р. при начислении простых процентов во всех трех случаях. Описать функцию вычисления суммы, применить ее для вектора из 30 сгенерированных различных значений  $X$ ,  $Y$ ,  $Z$ . Построить линейный график.

- 6.5.** В финансовом договоре клиента с банком предусмотрено погашение долга в размере  $X$  тыс. р. через  $Y$  дней при взятом кредите в 20 тыс. р. Определить доходность такой сделки для банка в виде годовой процентной ставки. При начислении банк использует простые обыкновенные проценты. Описать функцию вычисления суммы, применить ее для вектора из 30 сгенерированных различных значений  $X$ ,  $Y$ . Построить столбиковую диаграмму.
- 6.6.** Клиент поместил в банк вклад в сумме 100 тыс. р. под  $X\%$  годовых с ежеквартальной выплатой простых процентов. Какую сумму клиент будет получать каждый квартал? Описать функцию вычисления ежеквартальной суммы, применить ее для вектора из значений от 4% до 21%. Построить столбиковую диаграмму.
- 6.7.** Предприниматель взял в банке ссуду на два года под процентную ставку  $X\%$  годовых. Определить, во сколько раз сумма долга к концу срока ссуды будет больше выданной банком суммы, если банк начисляет простые проценты. Описать функцию вычисления суммы, применить ее для вектора из 30 сгенерированных различных значений  $X$ . Построить линейный график.
- 6.8.** Банк выдал ссуду на  $X$  дней в размере 10 тыс. р. под простую процентную ставку 30% годовых. Рассчитать доход банка, если при начислении простых процентов считается, что в году 360 дней. Описать функцию вычисления дохода, применить ее для вектора из 30 сгенерированных значений  $X$ . Построить столбиковую диаграмму.
- 6.9.** Найти величину дохода кредитора, если за предоставление в долг на полгода некоторой суммы он получил  $X$  тыс. р. При этом применялась простая процентная ставка в 22%. Описать функцию вычисления дохода, применить ее для вектора из 30 генерированных значений  $X$ . Построить столбиковую диаграмму.
- 6.10.** Клиент поместил в банк вклад  $X$  тыс. р. под простую процентную ставку 15% годовых. Какая сумма будет на счете клиента через 1–3 года? Описать функцию вычисления суммы, применить ее для вектора из 30 сгенерированных значений  $X$ . Построить линейный график.

- 6.11.** Банк принимает депозиты на 3 месяца по процентной ставке 18% годовых, на 6 месяцев – 24% годовых и на год – по 29% годовых. Определить сумму, которую получит владелец депозита в размере  $X$  тыс. р. при начислении простых процентов во всех трех случаях. Описать функцию вычисления суммы, применить ее для вектора из 30 сгенерированных значений  $X$ . Построить столбиковую диаграмму.
- 6.12.** Предприниматель взял в банке ссуду на  $X$  месяцев под процентную ставку 15% годовых. Определить, во сколько раз сумма долга к концу срока ссуды будет больше выданной банком суммы, если банк начисляет простые проценты. Описать функцию вычисления суммы, применить ее для вектора из различных значений  $X$  от 2 до 18. Построить линейный график.
- 6.13.** Банк выдал ссуду на 45 дней в размере  $X$  тыс. р. под простую процентную ставку 27% годовых. Рассчитать доход банка, если при начислении простых процентов считается, что в году 365 дней. Описать функцию вычисления дохода, применить ее для вектора из 30 сгенерированных значений  $X$ . Построить столбиковую диаграмму.
- 6.14.** Клиент поместил в банк вклад 100 тыс. р. под простую процентную ставку  $X\%$  годовых. Какая сумма будет на счете клиента через 1–12 месяцев? Описать функцию вычисления суммы, применить ее для вектора из значений от 4% до 25%. Построить столбиковую диаграмму.
- 6.15.** Найти величину дохода кредитора, если за предоставление в долг на полгода некоторой суммы денег он получил 45 тыс. р. При этом применялась простая процентная ставка в  $X\%$ . Описать функцию вычисления дохода, применить ее для вектора из значений  $X$  от 12% до 30%. Построить столбиковую диаграмму.

- 6.16.** В финансовом договоре клиента с банком предусмотрено погашение долга в размере  $X$  тыс. р. через 150 дней при взятом кредите в  $Y$  тыс. р. Определите доходность такой сделки для банка в виде годовой процентной ставки. При начислении банк использует простые обыкновенные проценты. Описать функцию вычисления суммы, применить ее для вектора из 30 сгенерированных различных значений  $X$ ,  $Y$ . Построить линейный график.
- 6.17.** Клиент поместил в банк вклад 200 тыс. р. под простую процентную ставку  $X\%$  годовых. Какая сумма будет на счете клиента через 1–3 года? Описать функцию вычисления суммы, применить ее для вектора из значений от 5% до 21%. Построить столбиковую диаграмму.
- 6.18.** Банк выдал ссуду на 45 дней в размере 20 тыс. р. под простую процентную ставку  $X\%$  годовых. Рассчитайте доход банка, если при начислении простых процентов считается, что в году 360 дней. Описать функцию вычисления дохода, применить ее для вектора из значений  $X$  от 12% до 32%. Построить столбиковую диаграмму.
- 6.19.** Банк выдал ссуду на 53 дня в размере  $P$  тыс. р. под простую процентную ставку 18% годовых. Рассчитать доход банка, если при начислении простых процентов считается, что в году 365 дней. Описать функцию вычисления дохода, применить ее для вектора из 15 сгенерированных значений  $P$ . Построить линейный график.
- 6.20.** Найти величину дохода кредитора, если за предоставление в долг на год некоторой суммы он получил  $P$  тыс. р. При этом применялась простая процентная ставка в 19%. Описать функцию вычисления дохода, применить ее для вектора из 15 генерированных значений  $P$ . Построить линейный график.



## ЛАБОРАТОРНАЯ РАБОТА №7

### *Вычисление интегралов и производных. Таблицы.*

#### *Интегрирование и дифференцирование*

Для вычисления интегралов по бесконечным и конечным пределам используется формула

```
integrate (f, lower, upper, ..., subdivisions = 100,  
rel.tol = .Machine$double.eps^0.25, abs.tol = rel.tol,  
stop.on.error = TRUE, keep.xy = FALSE, aux = NULL)
```

где  $f$  – функция, интегрируемая по первому аргументу;

$lower$  и  $upper$  – нижний и верхний пределы интегрирования, может быть и  $Inf$ ;

... – дополнительные аргументы функции  $f$ ;

$subdivisions$  – максимальное число интервалов интегрирования;

$rel.tol$  – требуемая относительная точность;

$abs.tol$  – требуемая абсолютная точность;

$stop.on.error$  – логический аргумент предотвращения ошибки;

$keep.xy$  и  $aux$  – не используются, введены для совмещения с языком S.

**Пример 1.** Вычислить интеграл от функции  $f(x) = \frac{1}{(x+1)\sqrt{x}}$ .

Предел интегрирования от 0 до  $\infty$ .

```
integrand = function(x) {1/((x+1)*sqrt(x))}  
integrate(integrand, lower = 0, upper = Inf)
```

**Пример 2.** Интегрируемая функция с двумя параметрами

$$f(x) = \frac{1}{a(1+(x-b)^2)\sqrt{x}}, a = 2, b = 2.$$

```
> integrand=function(x,a,b) {1/(a*(1+(x-b)^2))}  
> integrate(integrand, lower=0,upper=Inf,a=2,b=2)  
1.338973 with absolute error < 5.5e-08
```

Вычисление производных выполняется функциями:

$D(expr, name)$

$deriv(expr, namevec, function.arg = NULL, tag = “.expr”,  
hessian = FALSE, ..)$

```
deriv3(expr, namevec, function.arg = NULL, tag = ".expr",
hessian = TRUE, ..)
```

Функция `D()` вычисляет производные по одному аргументу, функции `deriv()` и `deriv3()` вычисляют частные производные.

Аргументы:

`expr` – любое выражение или функция;

`name`, `namevec` – символьный вектор, задающий имена переменных, по которым берутся производные,

`function.arg` – символьный вектор аргументов результирующей функции,

`tag` – префикс для обозначения локальных переменных при выводе результатов;

`hessian` – нужно ли вычислять вторые производные.

**Пример 3.** Вычислить первую и вторую производную функции  $f(x, y) = \sin(\cos(x + y^2))$  по переменным  $x$  и  $y$ . Затем вычислить значение найденных производных в точках  $x = \frac{\pi}{4}, y = \frac{\pi}{6}$ .

```
> trig = expression(sin(cos(x+y^2)))
> (D.x = D (trig, 'x'))
-(cos(cos(x + y^2)) * sin(x + y^2))
> (D.y = D (trig, 'y'))
-(cos(cos(x + y^2)) * (sin(x + y^2) * (2 * y)))
```

Соответственно вторые производные вычисляются функциями:

```
> (D.x.x = D (D.x, 'x'))
-(sin(cos(x + y^2)) * sin(x + y^2) * sin(x + y^2) +
  cos(cos(x + y^2)) * cos(x + y^2))
> (D.y.y = D (D.x, 'y'))
-(sin(cos(x + y^2)) * (sin(x + y^2)*(2 * y))*sin(x + y^2) +
  cos(cos(x + y^2)) * (cos(x + y^2) * (2 * y)))
> (D.y.y = D (D.y, 'y'))
-(sin(cos(x + y^2)) * (sin(x + y^2) * (2*y))*(sin(x+y^2) *
  (2 * y)) + cos(cos(x + y^2)) * (cos(x+y^2) * (2 * y) *
  (2 * y) + sin(x + y^2) * 2))
```

```
> (D.y.x = D (D.y, 'y'))
-(sin(cos(x + y^2)) * (sin(x + y^2) * (2*y))*(sin(x+y^2) *
  (2 * y)) + cos(cos(x + y^2)) * (cos(x+y^2)*(2*y) *
  (2 * y) + sin(x + y^2) * 2))
```

Для поиска значений найденных производных в заданных точках можно использовать функцию `eval()`.

```
> x=pi/4; y=pi/6
> eval(D.x)
[1] -0.7698184
> eval(D.y)
[1] -0.8061519
> eval(D.x.x)
[1] -0.7893344
> eval(D.y.y)
[1] -2.405239
> eval(D.y.x)
[1] -2.405239
```

При помощи функций `derive()` и `deriv3()` можно вычислить частные производные.

```
> f = expression(x^2 + 5*x + 1)
> print(deriv(f, "x"))
expression({
  .value <- x^2 + 5 * x + 1
  .grad <- array(0, c(length(.value), 1L), list(NULL, c(
"x"))))
  .grad[, "x"] <- 2 * x + 5
  attr(.value, "gradient") <- .grad
  .value
})

> print(deriv(quote(sinpi(x^2)), "x"))
expression({
  .expr1 <- x^2
  .value <- sinpi(.expr1)
  .grad <- array(0, c(length(.value), 1L), list(NULL, c(
"x"))))
  .grad[, "x"] <- cospi(.expr1) * (pi * (2 * x))
```

```

    attr(.value, "gradient") <- .grad
    .value
  })

> deriv(trig, c('x', 'y'), func = TRUE)
function (x, y)
{
  .expr2 <- x + y^2
  .expr3 <- cos(.expr2)
  .expr5 <- cos(.expr3)
  .expr6 <- sin(.expr2)
  .value <- sin(.expr3)
  .grad <- array(0, c(length(.value), 2L), list(NULL, c(
"x",
  "y"))))
  .grad[, "x"] <- -(.expr5 * .expr6)
  .grad[, "y"] <- -(.expr5 * (.expr6 * (2 * y)))
  attr(.value, "gradient") <- .grad
  .value
}

```

Получен градиент — частные производные и `.value` — исходная (дифференцируемая) функция.

Для определения точки минимума / максимума на заданном интервале можно использовать функции `optimize()` и `optimise()`.

```

optimize(f= , interval = , ..., lower = min(interval), upper
= max(interval), maximum = FALSE, tol = .Machine$double.ep
s^0.25)
optimise(f= , interval = , ..., lower = min(interval), upper
= max(interval), maximum = FALSE, tol = .Machine$double.ep
s^0.25)

```

Аргументы:

`f` — оптимизируемая функция;

`interval` — числовой вектор, задающий интервал на котором ищется экстремум;

`...` — дополнительные аргументы функции `f`;

`lower` — нижняя граница интервала оптимизации;

`upper` — нижняя граница интервала оптимизации;

maximum – нужно искать минимум (по умолчанию) или максимум функции;  
tol – требуемая точность.

**Пример 4.** Найти минимум и максимум функции  $f(x) = \frac{(x-5)^2 \sin(x)}{(x+2)^2}$  на интервалах  $[-1; 4]$ ,  $[0; 3]$ ,  $[1; 3]$ .

```
> f=function(x,a,b) {(x-a)^2*sin(x)/((x+b)^2)}
> xmin=optimize(f, c(-1,4),tol = 0.0001,a=5,b=2)
> xmin
$minimum
[1] 3.642265

$objective
[1] -0.02779574

> xmax=optimize(f, c(-1,4),tol = 0.0001,a=5,b=2,maximum=T)
> xmax
$maximum
[1] 0.691799

$objective
[1] 1.634088

> xmin=optimize(f, c(0,3),tol = 0.0001,a=5,b=2)
> xmin
$minimum
[1] 2.999958

$objective
[1] 0.02258725

> xmax=optimize(f, c(0,3),tol = 0.0001,a=5,b=2,maximum=T)
> xmax
$maximum
[1] 0.6917868

$objective
[1] 1.634088

> xmin=optimize(f, c(1,3),tol = 0.0001,a=5,b=2)
```

```

> xmin
$minimum
[1] 2.999952

$objective
[1] 0.02258839

> xmax=optimise(f, c(1,3),tol = 0.0001,a=5,b=2,maximum=T)
> xmax
$maximum
[1] 1.000048

$objective
[1] 1.49591

```

Функция `nlm()` использует метод Ньютона для поиска минимума функции, частные и вторые производные.

```

nlm(f, p, ..., hessian = FALSE, typesize = rep(1, length(p))
, fscale = 1, print.level = 0, ndigit = 12, gradtol = 1e-6
, stepmax = max(1000*sqrt(sum((p/typesize)^2)),1000), step
tol = 1e-6, iterlim = 100, check.analyticals = TRUE)

```

Основные аргументы:

`f` – минимизируемая функция;  
`p` – начальные значения для аргументов, по которым ищется минимум;  
`...` – дополнительные параметры для функции `f`.

Результаты работы функции `nlm()`:

`minimum` – значение функции `f` в точке минимума;  
`estimate` – точка, в которой достигается минимум функции;  
`gradient` – значение градиента в найденной точке минимума;  
`hessian` – матрица вторых производных;  
`code` – код завершения: 1 и 2 – значения скорее всего являются решением, 3–5 – сообщения об ошибках;  
`iterations` – число выполненных итераций.

**Пример 5.** Найти минимум и максимум функции  $f(x) = \frac{(x-5)^2 \sin(x)}{(x+2)^2}$  на интервалах  $[-1; 4]$ ,  $[0; 3]$ ,  $[1; 3]$  (используя метод Ньютона).

```
> f=function(x,a,b) {(x-a)^2*sin(x)/((x+b)^2)}  
> nlm(f,0,a=5,b=2)  
$minimum  
[1] -8.076856e+14
```

```
$estimate  
[1] -2
```

```
$gradient  
[1] 3.966926e+20
```

```
$code  
[1] 2
```

```
$iterations  
[1] 17
```

Зададим другое начальное значение параметра  $x$ .

```
> f=function(x,a,b) {(x-a)^2*sin(x)/((x+b)^2)}  
> nlm(f, 4, a=5,b=2)  
$minimum  
[1] -0.02779574
```

```
$estimate  
[1] 3.642274
```

```
$gradient  
[1] 4.607483e-09
```

```
$code  
[1] 1
```

```
$iterations  
[1] 7
```

При разных начальных значениях можно получить различные точки минимума.

## Таблицы

Таблицы или фреймы данных являются одним из основных типов языка. В отличие от матриц таблицы могут в разных столбцах содержать данные разных типов, но в столбце находятся данные одного типа. Таблица создается с помощью функции `data.frame()`.

### Пример 6. Таблица – список студентов

```
> data1<-c(1:5)
> data2<-c('Petr','Alex','Olga','Bob','Nik')
> fr<-data.frame(data1, data2)
> fr
```

	data1	data2
1	1	Petr
2	2	Alex
3	3	Olga
4	4	Bob
5	5	Nik

Доступ к элементам таблицы может быть матричный – указывается номер строки и столбца, или списковый – указывается номер или имя строки, а затем имя или номер столбца.

Функция поиска в таблице `grep()` позволяет в заданном столбце (задается в формате `имя_таблицы$имя_столбца`) найти некоторое вхождение. В качестве результата указываются строки с этим вхождением.

Можно сформировать таблицу, содержащую только строки с указанным вхождением.

```
> mydata<-data.frame(name=c('Antony','Bob','Cecilia',
  'Jack','Mary','Tony'), col_a=c(0,1,0,1,2,3),
  col_b=c(3,0,3,3,1,0), stringAsFactors=FALSE)
> mydata
```

	name	col_a	col_b	stringAsFactors
1	Antony	0	3	FALSE
2	Bob	1	0	FALSE
3	Cecilia	0	3	FALSE
4	Jack	1	3	FALSE



```

5   Mary      2      1      FALSE
6   Tony      3      0      FALSE
> grep('[t,T]ony', mydata$name)
[1] 1 6
> mydata[grep('[t,T]ony', mydata$name),]
  name col_a col_b stringAsFactors
1 Antony    0     3          FALSE
6  Tony     3     0          FALSE

```

Другой способ отобрать строки таблицы с заданным значением в некотором столбце:

```

> mydata<-data.frame(name=c('Antony','Bob','Cecilia','Jack',
', 'Mary','Tony'), col_a=c(0,1,0,1,2,3), col_b=c(3,0,3,3,1,
0), stringAsFactors=FALSE)
> mydata
  name col_a col_b stringAsFactors
1 Antony    0     3          FALSE
2   Bob     1     0          FALSE
3 Cecilia    0     3          FALSE
4  Jack     1     3          FALSE
5  Mary     2     1          FALSE
6  Tony     3     0          FALSE
> h=mydata[which(mydata$col_a==1),]
> h
  name col_a col_b stringAsFactors
2  Bob     1     0          FALSE
4 Jack     1     3          FALSE

```

Для конкретного значения некоторого столбца можно посчитать стандартную функцию (aggregate), например, среднее арифметическое (mean):

```

> mydata<-data.frame(country=c('France','Italy','France','
Italy','Italy','Russia'), col_a=c(0,1,0,1,2,3), col_b=c(3
,0,3,3,1,0), stringAsFactors=FALSE)
> mydata
  country col_a col_b stringAsFactors
1  France     0     3             FALSE
2   Italy     1     0             FALSE
3  France     0     3             FALSE
4   Italy     1     3             FALSE
5   Italy     2     1             FALSE
6  Russia     3     0             FALSE
> aggregate(mydata[, 'col_a'],list(mydata$country),mean)
  Group.1      x
1  France 0.000000
2   Italy 1.333333
3  Russia 3.000000

```

### ***Задание***

1. В соответствии с таблицей исходных данных построить таблицу данных (не менее 4 столбцов, имеющих данные разных типов: числовые, строковые). Сделать выборку строковых данных по трем различным запросам на столбцы. Для числовых столбцов вычислить сумму значений, среднее арифметическое, величину заданного процента.
2. В соответствии с таблицей исходных данных необходимо построить гистограмму и столбиковую диаграмму по столбцам из данных сгенерированной таблицы (например, по типам билетов, по стоимости).
3. Используя функцию `curve()`, построить два графика функций, отражающих суммарные показатели, например, по дням (декадам, месяцам) в соответствии Таблицей исходных данных. Подписи осей и наличие заголовков графиков обязательны.
4. Выполнить индивидуальное задание.

**Индивидуальное задание (интегралы, производные)**

Вычислить следующие интегралы:

$$\begin{array}{ll} 7.1. \int \frac{\sqrt{x^4+x^{-4}+2}}{x^3} dx & 7.2. \int \frac{x^2+3}{x^2-1} dx \\ 7.3. \int \frac{2^{x+1}+5^{x-1}}{10^x} dx & 7.4. \int \frac{1-x^7}{x(1+x^7)} dx \end{array}$$

Найти производные следующих функций:

7.5.  $f(x) = x^3 + 3x^2 - 72x + 90$  в точке  $x = 5$ .

7.6.  $f(x) = \frac{x^2}{x+2}$  в точке  $x = -4$ .

7.7.  $f(x) = (x^2 - 2x + 3)^5$

7.8.  $f(x) = \sqrt{x}$  в точке  $x = 4$ .

7.9.  $f(x) = x^3 \sin x + 3x^2 \cos x - 6 \sin x - 6 \cos x$ .

7.10. Найти вторую производную функции  $f(x) = \sin^2 \frac{x}{3}$ .

7.11. Найти  $f''(0)$  для функции  $f(x) = \frac{x}{\sqrt{x^2+1}}$ .

Вычислить следующие интегралы:

$$7.12. \int_{-1}^1 \frac{1}{1+x^2} dx. \quad 7.13. \int_{-1}^{\sqrt{3}} \frac{1}{1+x^2} dx.$$

$$7.14. \int_0^{\pi/4} \frac{\sin x}{\cos^2 x} dx. \quad 7.15. \int_0^{\pi/4} \frac{dx}{1+x}$$

Найти экстремумы функций:

7.16.  $y = 7 + 12x - x^3$ . 7.17.  $y = x - 2\sqrt{x-2}$ .

7.18.  $y = 3 - 22x + x^4$ . 7.19.  $y = 2x + 3\sqrt{x-1}$ .

7.20. Найти  $f''(0)$  для функции  $f(x) = \frac{2x}{\sqrt{x^2+3}}$ .

## ЛАБОРАТОРНАЯ РАБОТА №8

### *Датасеты. Исследование характеристик объектов индивидуальной тематики*

Для выполнения работы с датасетом понадобится подключить следующие пакеты:

```
library("psych")# описательные статистики  
library("ggplot2")# графики  
library("dplyr") # манипуляции с данными
```

Получим описание набора данных по автомобилям cars командой:

```
help(cars)
```

Поместим в переменную d встроенный в R набор данных по автомобилям:

```
d <- cars # этот набор данных находится в базовом пакете datasets
```

Теперь d имеет тип данных data.frame (набор данных),

Следующей командой можно посмотреть на этот набор данных, в результате чего будут перечислены все переменные и типы данных:

```
glimpse(d) # функция из пакета dplyr
```

Результат выполнения команды появится в консоли:

```
> d<-cars  
> glimpse(d)  
Rows: 50  
Columns: 2  
$ speed <dbl> 4, 4, 7, 7, 8, 9, 10, 10, 10, 11, 11, 12,  
12, 12, 12, 13, 13, 13, 13, 14, 14, 14, 14~  
$ dist <dbl> 2, 10, 4, 22, 16, 10, 18, 26, 34, 17, 28,  
14, 20, 24, 28, 26, 34, 34, 46, 26, 36, 60~
```

Переменные speed и dist имеют тип данных dbl (double) и содержат по 50 наблюдений.

Посмотрим на первые шесть наблюдений набора данных командой head(d).

```
> d<-cars  
> glimpse(d)  
Rows: 50  
Columns: 2  
$ speed <dbl> 4, 4, 7, 7, 8, 9, 10, 10, 10, 11, 11, 12,  
12, 12, 12, 13, 13, 13, 13, 14, 14, 14, 14~
```

```
$ dist <dbl> 2, 10, 4, 22, 16, 10, 18, 26, 34, 17, 28,
14, 20, 24, 28, 26, 34, 34, 46, 26, 36, 60~
```

```
> head(d)
```

```
  speed dist
1     4     2
2     4    10
3     7     4
4     7    22
5     8    16
6     9    10
```

и последние шесть наблюдений: `tail(d)`

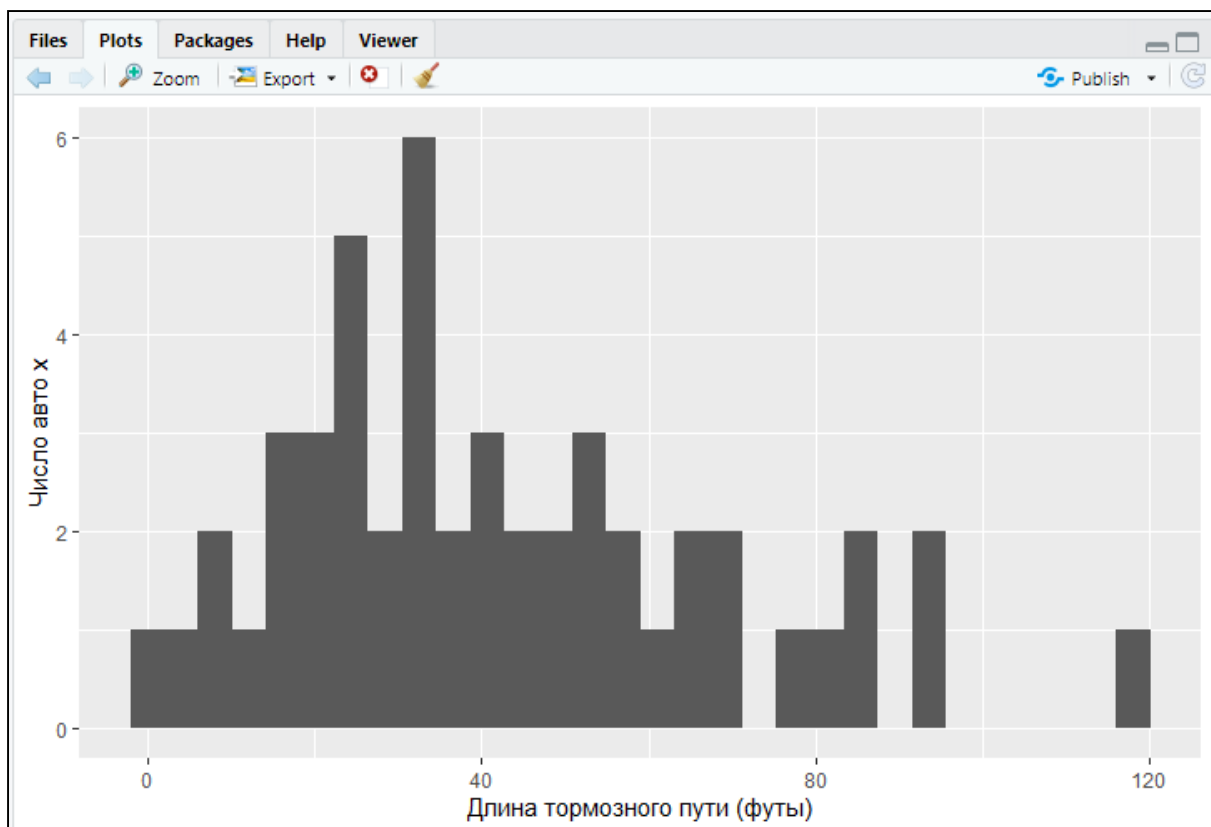
Получим таблицу с описательными статистиками: среднее, мода, медиана, стандартное отклонение, минимум / максимум, асимметрия, эксцесс и т.д.:

```
> describe(d)
```

```
vars  n  mean    sd median trimmed  mad min max
speed  1 50 15.40  5.29     15  15.47  5.93   4  25
dist   2 50 42.98 25.77     36  40.88 23.72   2 120
      range skew kurtosis  se
speed   21 -0.11   -0.67 0.75
dist   118  0.76    0.12 3.64
```

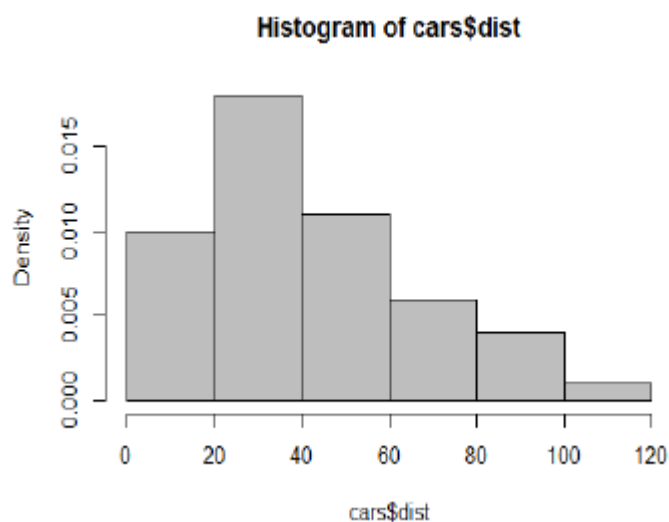
Построим гистограмму абсолютных частот для переменной (длины тормозного пути). Воспользуемся функцией `qplot`, задав источник данных `d` (аргумент `data`), переменную для построения графика (`dist`), подпишем оси (параметры функции `xlab` и `ylab`) и название графика (параметр `main`):

```
library("ggplot2")
# функция из пакета ggplot2
qplot(data = d, dist, xlab = "Длина тормозного пути (футы)",
ylab = "Число авто x")
```



Построим гистограмму плотности распределения (функция из базового пакета `graphics`):

```
hist(d$dist, probability = TRUE, col="grey")
```

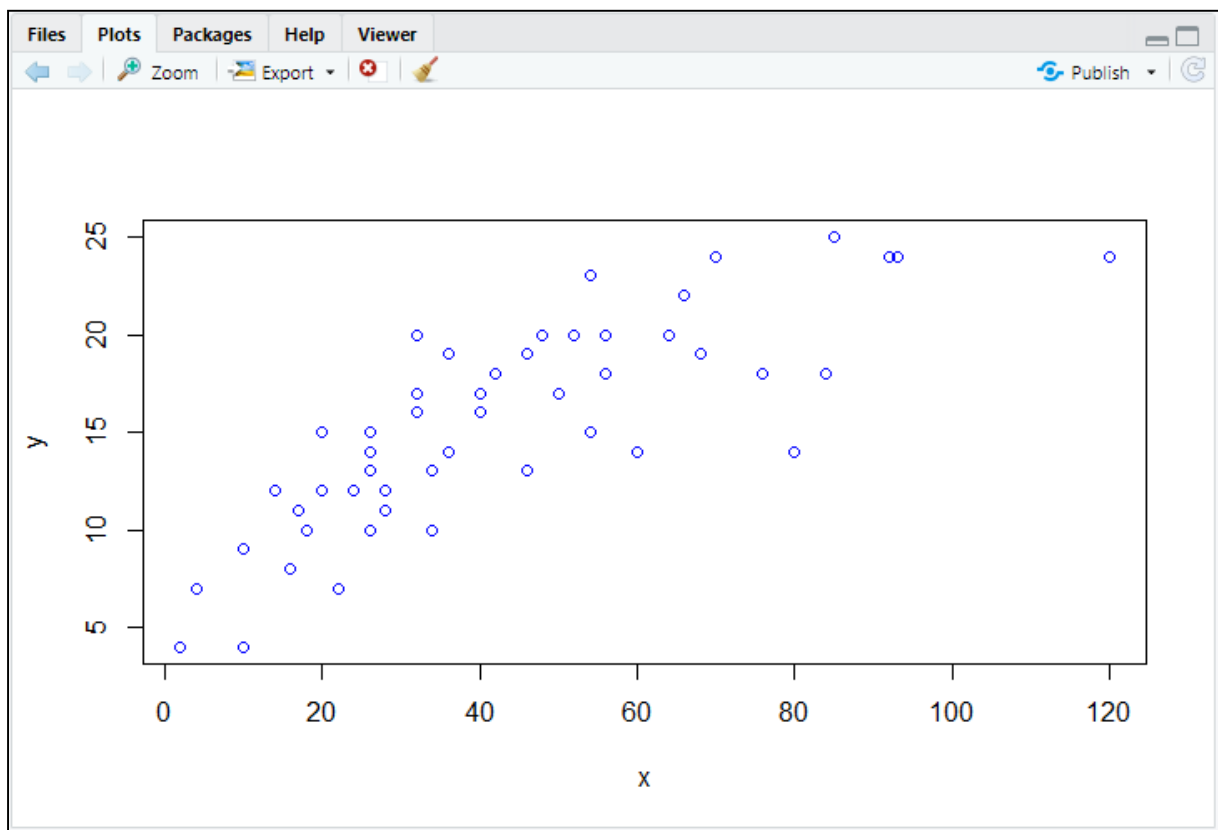


Построение простого графика на датасет:

```
x<-d$dist  
y<-d$speed  
plot(x,y,col="blue")
```

Весь фрагмент:

```
library("dplyr")  
library("psych")  
d<-cars  
glimpse(d)  
head(d)  
describe(d)  
x<-d$dist  
y<-d$speed  
plot(x,y,col="blue")
```



## *Исследование характеристик объектов индивидуальной тематики*

Для построения линейной регрессии создадим таблицу в MS Excel. Сохраняем файл в формате «csv» – firma.csv.

	A	B	C
1	№	Reclama	Sale
2	1	10	750
3	2	10	790
4	3	12	820
5	4	14	810
6	5	14	780
7	6	15	900
8	7	17	910
9	8	17	870
10	9	20	900
11	10	21	780
12	11	21	810

Далее считываем исходные данные в структуру linreg.

```
> getwd()
[1] "C:/Users/p000/Documents"
> linreg<-read.table('firma.csv', header = TRUE,
sep = ';')
> head(linreg)
  X. Reclama Sale
1  1      10  750
2  2      10  790
3  3      12  820
4  4      14  810
5  5      14  780
6  6      15  900
> str(linreg)
'data.frame': 20 obs. of  3 variables:
 $ X.      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Reclama: int  10 10 12 14 14 15 17 17 20 21 ...
 $ Sale   : int  750 790 820 810 780 900 910 870 900 780 ...
```



Диаграмма рассеяния введенных данных может быть построена:

```
> plot(linreg, cex = 2, main = 'Диаграмма
рассеяния')
```



Затем следует выбрать вид линии, которая будет описывать эту связь. В качестве такой линии возьмем прямую линию вида  $\text{Sale} = a + b \times \text{Reclama}$ .

В R основная функция для подгонки регрессионных моделей – это `lm()`.

Формат ее применения: `myreg <- lm(formula, data)`, где `formula` описывает вид модели, которую нужно подогнать, а `data` – это таблица с данными, которые используются для создания модели.

```
getwd()
linreg<-read.table('firma.csv', header = TRUE, sep = ';')
head(linreg)
str(linreg)
plot(linreg, cex = 2, main = 'Диаграмма рассеяния')
reg<-lm(Sale~Reclama, data = linreg)
summary(reg)
```

Результат работы этого фрагмента следующий:

```
> reg<-lm(Sale~Reclama, data = linreg)
> summary(reg)
```

Call:

```
lm(formula = Sale ~ Reclama, data = linreg)
```

Residuals:

Min	1Q	Median	3Q	Max
-80.681	-38.483	7.288	33.329	72.943

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	<b>742.995</b>	31.059	23.922	4.29e-15 ***
Reclama	<b>5.604</b>	1.420	3.946	<b>0.000946</b> ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 44.86 on 18 degrees of freedom

Multiple R-squared: 0.4639, Adjusted R-squared: 0.4341

F-statistic: 15.57 on 1 and 18 DF, p-value: **0.0009461**

Из полученных результатов следует, что уравнение для предсказания объема продаж от затрат на рекламу имеет следующий вид:

$$\text{Sale} = 742.995 + 5.604 \times \text{Reclama}.$$

Из столбца Pr(>|t|) ясно, что коэффициент регрессии (5.604) статистически значимо отличается от нуля ( $p < 0.001$ ) и означает, что на каждую тысячу условных единиц затрат на рекламу ожидается увеличение объемов затрат на 5.604 единицы.

Построим доверительные интервалы для коэффициентов регрессии с помощью функции confint() с уровнем значимости 0.05 (по умолчанию):

```
> confint(reg)
```

	2.5 %	97.5 %
(Intercept)	-538.22694	54.48500
Reclama	14.50752	41.00116

```
(Intercept) 677.742605 808.248178
Reclama      2.620742    8.587432
```

Коэффициент регрессии  $a$  попадает в интервал [677.742605, 808.248178], поэтому нулевую гипотезу по оценке коэффициента можно отбросить на уровне значимости 0.05, что подтверждается значением  $\Pr(>|t|) = 0.0009461$ .

Коэффициент регрессии  $b$  попадает в интервал [2.620742, 8.587432], поэтому нулевую гипотезу по оценке коэффициента можно отбросить на уровне значимости 0.05, что подтверждается значением  $\Pr(>|t|) = 0.0009461$ .

Выполним дисперсионный анализ подогнанной модели:

```
> anova(reg)
```

```
Analysis of Variance Table
```

```
Response: Sale
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Reclama	1	31349	31349.3	15.575	0.0009461 ***
Residuals	18	36231	2012.8		

```
---
```

```
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Для получения необходимого прогноза используем следующий программный код:

```
> nov<-data.frame(Reclama = 15)
> predict(reg,newdata = nov)
1
827.0567
```

Таким образом, при затратах на рекламу 15000 условных единиц ожидается увеличение продаж на 827 тыс. р. Построим доверительные интервалы для среднего "confidence" и индивидуального "prediction" прогнозных значений, выполняя следующий программный код:

```
> predict(reg,newdata = nov, interval = 'confidence')
      fit      lwr      upr
1 827.0567 799.9756 854.1379
> predict(reg,newdata = nov, interval = 'prediction')
      fit      lwr      upr
1 827.0567 728.9868 925.1267
```

### ***Задания***

1. Создать таблицу данных на основе индивидуальной тематики (см. приложение) из 20 строк и 4 переменных с помощью конструкции `data.frame()`. Среди переменных должна быть количественная переменная `F`, например `Стоимость`.

2. Некоторые переменные представить как факторные. Проверить, что эти переменные действительно факторные любым удобным способом с соответствующим сообщением в консоли.

3. Добавить в таблицу новую переменную `F30`, значения которой равны исходная стоимость + премия в размере 30% (или надбавка). Эта переменная может быть вычислена по любой формуле в соответствии с индивидуальной тематикой.

4. Рассчитать среднее значение количественных переменных, их медиану, максимальные и минимальные значения и вывести их на экран.

5. Используя инструменты `R`, построить столбчатую диаграмму для количественных переменных.

6. Добавить новую переменную `M`. Необходимо найти максимальное значение `Стоимости` (переменная `F`) и для каждой строки определить, сколько процентов от максимального значения составляет текущая `Стоимость`.

7. Рассчитать среднее значение переменной `F` в зависимости от факторных переменных (например, от вида объектов).

8. Выделить две количественные переменные, между которыми необходимо найти зависимость, например, при продаже кондиционеров: температура воздуха и объем продаж. С помощью регрессионного анализа выявить наличие или отсутствие зависимости. Построить диаграмму рассеяния.

9. Построить прогноз некоторой количественной переменной с помощью функции `predict()`.

### ***Индивидуальное задание (обработка датасета)***

Загрузить данные для своего варианта в переменную-вектор.

Получить справочную информацию по своим данным, просмотреть их содержимое, выполнить краткое описание в отчете.

- 8.1.** Создать новую переменную-вектор, значение элемента которого будет равно 1, если значение в исходном векторе больше среднего, и -1, если значение переменной меньше среднего, и 0, если значение равно среднему.
- 8.2.** Вывести описательные статистики.
- 8.3.** Построить графики абсолютных частот и плотности распределения.

Вариант	Набор данных	Имя переменной (вектора)
1	CO2	conc
2		uptake
3	ChickWeight	weight
4		Time
5	Orange	age
6		Circumference
7	airquality	Wind
8		Temp
9	faithful	eruptions
10		waiting
11	ability.cov	general picture
12		blocks maze
13	airmiles	time series
14	austres	Numbers
15	Formaldehyde	carb
16	Formaldehyde	optden
17	HairEyeColor	Hair Sex
18	ChickWeight	weight Diet
19	Orange	Tree age
20	HairEyeColor	Eye Sex

## **ПРИЛОЖЕНИЕ**

### **ВАРИАНТЫ ИНДИВИДУАЛЬНЫХ ТЕМАТИК**

1. Обменный пункт: виды валют, курсы валют за некоторый период, операции обмена (вид валюты, объем, тип операции: купля / продажа).
2. Ювелирный магазин: типы изделий, стоимость, журнал покупки изделий (типы изделия, суммарная стоимость за день).
3. Поликлиника: журнал учета пациентов: идентификатор (ФИО), возраст, пол, идентификатор заболевания, данные анализов: количество лейкоцитов, уровень холестерина.
4. Кондитерский магазин: виды конфет, стоимость, объем, журнал продажи: вид, объем, суммарная стоимость за день.
5. Парикмахерская: прайс услуг, кассовый журнал: тип услуги, объем продажи за день.
6. Школа, журнал успеваемости: предметы, баллы за полугодие.
7. Гостиницы: типы номеров, стоимости, журнал учета: количество сданных номеров, суммарная стоимость за сутки, стоимость по типам.
8. Книжный магазин: книги, стоимость, объем в страницах, регистрация поступлений и продаж.
9. Ремонтная мастерская: виды работ, стоимость, заказы на ремонт: суммарная стоимость, стоимость по видам, стоимость за период.
10. Аптечный киоск: номенклатура лекарств, стоимость, объемы, количество штук в упаковке, журнал регистрации продаж.
11. Столовая: виды блюд, стоимость, себестоимость продуктов, журнал заказов по дням.
12. Фотомастерская: прайс работ, журнал поступления заказов: виды работ, длительность, объем в штуках, стоимость.
13. Холдинг: журнал учета отчетных данных: список регионов, список предприятий, объем продукции по декадам, объем продаж по декадам, число работников за месяц.
14. Учет расхода материалов в компании: журнал учета расхода канцтоваров: виды канцтоваров, стоимость, объемы, количество штук в упаковке, учет расхода по декадам.

15. Сбор урожая: список видов продукции, журнал учета сбора урожая: вид продукции, объем по дням, число ящиков, стоимость продукции для продажи, заработная плата.
16. Экономическая база данных: отчетные статистические данные: регион, экономическая эффективность, количество занятых, количество самозанятых, количество безработных (помесячно).
17. Учет компьютерного оборудования: журнал регистрации выдачи оборудования: вид оборудования, стоимость, количество в упаковке, суммарная стоимость по видам и за период.
18. Начисление налогов в бюджет: журнал учета поступления налогов: предприятия, виды налогов, объемы налогов, доля задолженности, пеня.
19. Ремонтная мастерская электронного оборудования: журнал учета запасных частей: виды, стоимость, количество в упаковке, объем, затраты по декадам.
20. Магазин по продаже автомобилей: журнал поступления автомобилей: марка, стоимость, пробег, объем двигателя, наличие дополнительных опций, учет продаж по месяцам.
21. Система здравоохранения: журнал регистрации выдачи путевок в санатории: стоимость, тип путевки, длительность, количество процедур, суммарный объем продаж ежемесячно.
22. Туристические агентства: журнал регистрации продаж туров: идентификатор тура, количество городов или стран в туре, длительность, стоимость одноместного, двухместного размещения, суммарные объемы продаж ежемесячно.
23. Продажа билетов на рейсы: журнал продаж билетов: стоимость билета, тип билета, длительность маршрута в минутах, суммарные продажи по дням, по конечным пунктам.
24. Продажа пиломатериалов: журнал учета продаж пиломатериалов: виды пиломатериалов, габариты, стоимость, количество в партии, суммарные объемы продаж по видам, по декадам.
25. Склад металлоконструкций: журнал учета продаж: стоимость, количество в упаковке, вес единицы, суммарные продажи по дням и ежемесячно.

## РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. URL: <https://tsamsonov.github.io/r-geo-course/index.html>
2. Шипунов А.Б., Балдин Е.М., Волкова П.А. и др. Наглядная статистика. Используем R! М.: Форум, 2021.
3. Мастицкий С.Э., Шитиков В.К. Статистический анализ и визуализация данных с помощью R. М.: ДМК. Пресс, 2015.
4. Язык\_программирования\_R. URL: <https://ru.wikibooks.org/wiki>
5. Уланов В. А. Сборник задач по курсу финансовых вычислений / под ред. проф. В.В. Ковалева. М.: Финансы и статистика, 2000.



## СОДЕРЖАНИЕ

1. ОРГАНИЗАЦИЯ ЗАНЯТИЙ ПО КУРСУ .....	3
1.1. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ ОТЧЕТОВ О ВЫПОЛНЕНИИ ЛАБОРАТОРНЫХ РАБОТ .....	3
1.2. ГРАФИК ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ .....	4
1.3. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ОТЧЕТНОЙ ДОКУМЕНТАЦИИ....	4
2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ.....	6
ЛАБОРАТОРНАЯ РАБОТА №1 .....	6
ЛАБОРАТОРНАЯ РАБОТА №2 .....	10
ЛАБОРАТОРНАЯ РАБОТА №3 .....	15
<i>Индивидуальное задание (обработка вектора)</i> .....	20
ЛАБОРАТОРНАЯ РАБОТА №4 .....	23
<i>Индивидуальное задание (поиск максимума / минимума;     построение графика функции)</i> .....	32
ЛАБОРАТОРНАЯ РАБОТА №5 .....	34
<i>Индивидуальное задание (матрицы)</i> .....	40
ЛАБОРАТОРНАЯ РАБОТА №6 .....	48
<i>Индивидуальное задание (функции)</i> .....	52
ЛАБОРАТОРНАЯ РАБОТА №7 .....	56
<i>Индивидуальное задание (интегралы, производные)</i> .....	66
ЛАБОРАТОРНАЯ РАБОТА №8 .....	67
<i>Индивидуальное задание (обработка датасета)</i> .....	76
ПРИЛОЖЕНИЕ .....	77
РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА .....	79

*Учебное издание*

Добровольская Наталья Юрьевна  
Гаркуша Олег Васильевич

## **ЯЗЫК ПРОГРАММИРОВАНИЯ R**

Практикум

---

Подписано в печать 1.11.2024. Выход в свет 7.11.2024.  
Печать цифровая. Формат 60×84 <sup>1</sup>/<sub>16</sub>. Уч.-изд. л. 5,0.  
Тираж 500 экз. Заказ № 5853

Кубанский государственный университет  
350040, г. Краснодар, ул. Ставропольская, 149.

Издательско-полиграфический центр  
Кубанского государственного университета  
350040, г. Краснодар, ул. Ставропольская, 149.