# Практическая работа 2

# 1. Язык программирования R: базовые операции

### 1.1 Язык R

Язык R представляет собой мощный инструмент для статистики, графики и статистического программирования. Этот язык широко используется для проведения глубокого статистического анализа. Это бесплатная система с открытым исходным кодом, реализация которой является коллективным достижением большого числа людей. Для этого языка существует свыше 10 000 доступных пакетов дополнений. R является серьезным конкурентом всех коммерческих пакетов для статической обработки данных.

### 1.2 Загрузка и установка R

Полезные ссылки:

* Сайт R: https://cran.r-project.org/.
* Сайт RStudio: https://www.rstudio.com/.

Версии R часто обновляются. Данное руководство написано на примере версии 4.0.2.

**R для Windows:**

1. Кликнуть по ссылке *Download R for Windows*.
2. Выбирать *base*, далее *Download R 4.0.2 for Windows*
3. Скачать установочный файл с расширением .exe
4. Запустить файл и следовать инструкции по установке.

На шаге выбора папки для установки R, по возможности лучше выбрать папку, в названии которой нет текста на кириллице. Если на кириллице задано имя пользователя, можно сохранить R просто на диске С:.

**R для Mac OS**

1. Кликнуть по ссылке *Download R for (Mac) OS X*.

2. Кликнуть *R-4.0.2.pkg*.

3. Скачать установочный файл с расширением .pkg.

4. Запустить файл и следовать инструкции по установке.

**Установка RStudio**

1. Открыть сайт *https://www.rstudio.com*.
2. Выбрать в меню *Products - RStudio*.
3. Выбрать версию *RStudio Desktop*.
4. Кликнуть по *Download RStudio Desktop*.
5. Находим RStudio Desktop Open Source License и скачиваем установочный файл для своей системы (.exe для Windows, .pkg для Mac OS).
6. Загрузить файл и следовать инструкциям по установке.
7. Во время установки на Windows рекомендуется не отказываться от создания папок в меню и ярлыков.

**Облачный сервис RStudio**

*RStudio Cloud* – современный облачный сервис, который позволяет работать в RStudio, не устанавливая его на свой компьютер. Бесплатная регистрация, есть возможность создавать групповые проекты и делиться файлами.

Ссылка <https://rstudio.cloud/>.

### 1.3 Интерфейс RStudio

При запуске RStudio открывается рабочая область. Она состоит из нескольких окон, главное из них — это окно с консолью (здесь указана версия программы).

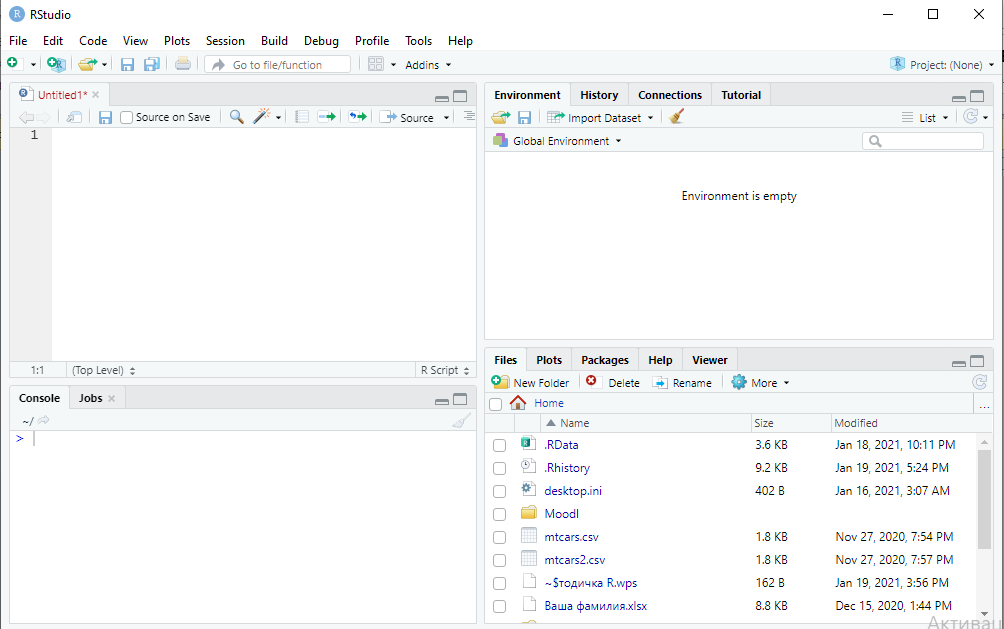


Рисунок 1.1 Окно RStudio

Верхний левый квадрант является редактором. Здесь вводится R-код, который можно сохранить для дальнейшего использования — функции, классы, пакеты и т. Д. Это, по сути, идентично главному окну любого другого редактора кода.

Верхний правый квадрант содержит вкладки: **Environment** (Окружение) и **History** (История). Вкладка Environment **отображает все созданные пользователем переменные, наборы данных, функции**. Для очистки содержимого окна Environment используется инструмент метелка.

Вкладка **History** служит для просмотра истории выполненных ранее команд. Команды отображаются в порядке их выполнения (недавние команды - внизу списка) и сгруппированы по времени.

Нижний правый квадрант содержит вкладки: **Files (Файлы), Plots (Графики), Packages (Пакеты), Help (**Справка**). Вкладка Files**, показывает содержимое рабочей директории. Вкладка **Plots** будет содержать графики, которые созданы с помощью R. На ней можно графики масштабировать, экспортировать, настраивать и проверять. Вкладка **Packages** позволяет устанавливать дополнительные пакеты в R. Краткое описание находится рядом с каждым доступным пакетом, хотя их намного больше, чем перечисленных там. Вкладка **Help** выдает справочную информацию о командах языка. Любые графики, которые вы генерируете, будут отображаться в панели **Plots** в правом нижнем углу.

Панель слева **Console** (*Консоль*). При запуске RStudio, она будет иметь информационный текст в верхней части консоли, текущую версию R. Ниже находится приглашение (командная строка). Как следует из названия, это по сути, запрос команды, где происходит взаимодействие с R. В консоль можно вводить код и сразу его исполнять. Расположение панелей можно изменить при помощи меню **View – Panes**.

## 1.4 Пакет R и скрипт R

Что такое пакет R и скрипт R? Это два ключевых компонента в языке.

**Пакет** — это набор R функций, файлов со справочной информацией и примерами, собранных вместе в одном архиве.  R пакеты играют важную роль, так как они используются как дополнительные расширения на базе R. Каждый пакет, как правило, посвящен конкретной теме, например: пакет 'ggplot2' используется для построения красивых векторных графиков.

R — это язык программирования с открытым кодом. В интернете в открытом доступе есть десятки тысяч пакетов R. Эти пакеты собраны в центральном репозитории. Вообще есть разные репозитории. Это и CRAN, и Bioconductor, и Github. Хранилище CRAN заслуживает отдельного упоминания – это сеть серверов, которые хранят большое количество версий кода и документации на R. Установить необходимый пакет можно командой **install.packages()**, например, **install.packages(“***ggplot2***”)** или при помощи диалогового окна

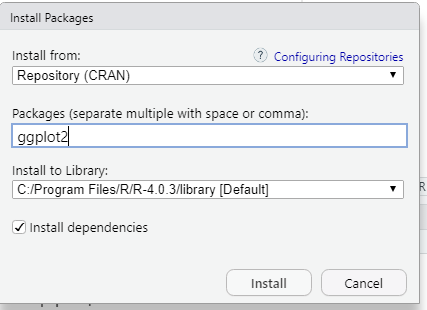


Рисунок 1.2 Диалоговое окно Install Packages

**R Script** — это текстовый файл с расширением. R, например мы может назвать скрипт *lesson*.R. Для создания R скрипта необходимо дать команды *File - New file – R script*. Или нажать комбинацию клавиш *Ctrl+Shift+N*. Откроется окно для написание программного кода.

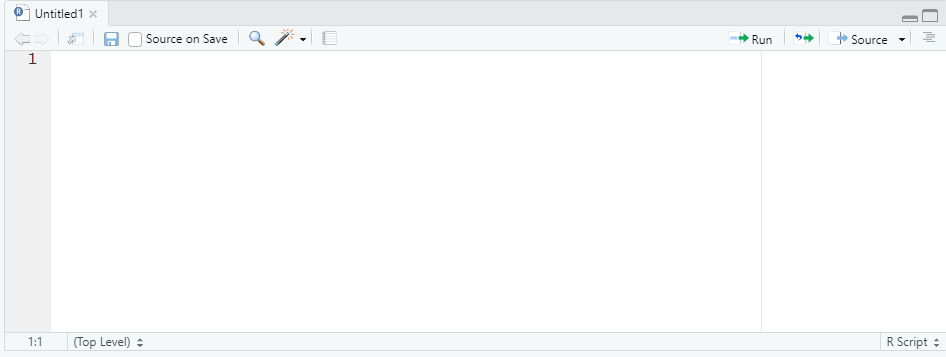


Рисунок 1.3 Окно для создания R-скрипта

Созданный скрипт можно сохранить в рабочей папке или в любом другом месте на компьютере. Рабочей папкой является папка, из которой по умолчанию запускается R studio. Если мы открываем файл с кодом, то рабочей папкой выступает папка, в которой этот файл лежит, то есть по умолчанию R studio видит только те файлы и папки, которые хранятся в рабочей папке.

Узнать, какая папка является рабочей на вашем компьютере можно с помощью команды ***getwd()***.

Можно сделать рабочей папку по своему желанию с помощью команды ***setwd(“путь”)***.

**Сохранение R скрипта**. Чтобы сохранить программу, необходимо дать команды ***File – Save as…*** Откроется окно для выбора кодировки (рисунок 1.4) текста программы. Для операционной системы Windows необходимо выбрать кодировку UTF-8, для операционной системы Mac – CP1251. Нажать Ок, и затем задать имя файлу.

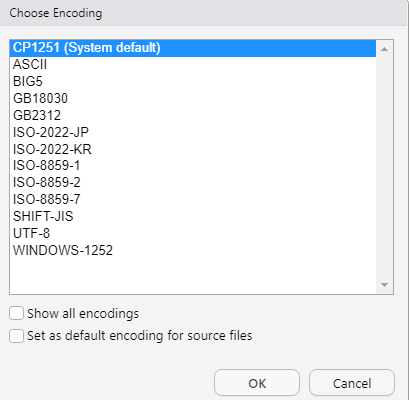


Рисунок 1.4 Окно выбор кодировки

R – объектно-ориентированный язык: переменные, данные, матрицы, функции, результаты, и т.д., хранятся в оперативной памяти компьютера в форме объектов, которые имеют имя. Язык R является интерпретируемым. Это значит, можно написать небольшие блоки кода и сразу его исполнить. Это отличает его от компилированных языков (например, C++).

Знак **>** в консоли - это приглашение к вводу команд, т.е. командная строка. Прежде, чем начать вводить команды, отметим общие пожелания по стилю кода:

1. Стоит избегать пробелов перед скобками, в функциях, например, which(), а не which ().
2. Вокруг операторов пробелы лучше добавлять. Например, вокруг +, /, >= и так далее.

На исполнение кода эти правила не влияют, но код выглядит аккуратнее и приятнее. В R есть свои стандарты оформления кода, можно почитать здесь: <http://adv-r.had.co.nz/Style.html>.

### 1.5 Базовые вычисления в R

Рассмотрим работу с числовыми переменными. Можно просто напечатать выражение, R вычислит, и результат будет отображен на экране (в панели *Console*), но не сохранен в памяти.

Чтобы запустить код на выполнение нужно сделать строку активной и нажать кнопку **Run** или нажать комбинацию клавиш <Ctrl>+ <Enter>.

Рассмотрим примеры вычислений.

(5 + 10) \* 2

[1] 30

48/8

[1] 6

48/7

[1] 6.857143

Получили дробный результат. Десятичным разделителем в дробных числах выступает точка, а не запятая.

Вычисление степени в R осуществляется двумя способами: при помощи **\*\***, либо **^**.

2\*\*3

[1] 8

или

2^3

[1] 8

Возвести число е в степень 1 – е1:

exp(1)

[1] 2.718282

e2:

exp(2)

[1] 7.389056

Если нужен логарифм по произвольному основанию, (логарифмирование данных очень часто используется для анализа данных), то используем функцию log. Основание логарифма указывается с помощью аргумента base.

Вычислим log10100.

log (100, base = 10)

[1] 2

Можно название аргумента опустить:

log (100, 10)

[1] 2

Если интересует натуральный логарифм – т.е. логарифм по основанию е, то основание можно опустить и использовать log без опций

log (100)

[1] 4.60517

Десятичные логарифмы вычисляются с помощью специальной функции *log10*.

log10(100)

[1] 2

Округление чисел в R выполняется с помощью функции:

*round(число, количество знаков после точки)*.

По умолчанию округление идет до целого.

Округлим до целого число 1.78.

round(1.78)

[1] 2

Если нужно округлить до десятых, необходимо добавить аргумент - количество знаков:

round(1.78, 1)

[1] 1.8

Для округления в меньшую сторону, используется функция:

*floor(число, кол-во знаков)*

floor(1.4)

[1] 1

Для округления в большую сторону используется функция:

*ceiling(число, кол-во знаков)*

ceiling(1.46)

[1] 2

В целом все операции с числами в R выглядят достаточно логично, и что важно, для их использования не нужно импортировать отдельные библиотеки.

### 1.6 Переменные в R

Переменные в R – это контейнер для хранения числа, текстовой строки, списка и т.д. В R строчные и заглавные буквы различаются, т.е. переменная с именем “x” и с именем “X” это разные переменные. Названия переменных в R могут содержать буквы, цифры, точки и знаки подчеркивания, при этом название переменной не может начинаться с цифры. Название переменной не должно совпадать со служебными словами (операторами) в R:if, else, for, while и другими.

Создадим переменную a и запишем в нее число 5:

a <- 5

a

[1] 5

**<-** оператор присваивания. Символ **<** пишется вместе со знаком минус так, чтобы они представляли стрелку, которая может быть направлена слева направо, либо наоборот:

a -> 5

a

[1] 5

Значение также может быть результатом арифметического выражения:

a <- 10+5

a

[1] 15

Можно присвоить и так:

a = 5

a

[1] 5

Такая команда также выполнится, но лучше использовать <-. Знак = в R используется в логических выражениях.

Как только создали переменную, она появляется в окне *Environment*. В этом окне отображаются все объекты, которые мы создаем во время работы.

В переменную v запишем последовательность чисел от 5 до 40:



Число в квадратных скобках перед результатом означает номер первого элемента в строке вывода. В данном примере это 5 под номером 1, и 31 под номером 27. Нумерация в R начинается с 1, а не с 0.

Текст в R заключается в кавычки – двойные или одинарные. Присвоим переменной hello значение welcome.

hello <- “welcome”

Вывести содержимое переменной hello:

hello

[1] “welcome”

## 1.7 Выражения в R

Для создания выражений в R могут быть использованы арифметические операторы, операторы сравнения, математические функции.

Арифметические операторы

|  |  |
| --- | --- |
| + | сложение |
| - | вычитание |
| \* | умножение |
| / | деление |
| ^ | возведение в степень |
| %% | остаток от деления |
| %//% | целая часть от деления |

Операторы сравнения, которые возвращают переменную логического типа (ИСТИНА или ЛОЖЬ)

|  |  |
| --- | --- |
| < | меньше |
| > | больше |
| <= | меньше или равно |
| >= | больше или равно |
| = = | равно |
| ! | отличный от |

Математические функции

|  |  |  |
| --- | --- | --- |
| Запись на R | Описание | Пример |
| pi | Константа π≈3,14 | pi 3.141593 |
| max(x,y) | Наибольшее число. Количество аргументов любое | max(1,4,5,1,2) 5 |
| min(x,y) | Наименьшее число. Количество аргументов любое | min(1,4,2,1,5) 1 |
| abs(x) | Модуль числа x | abs(-10) 10 |
| sqrt(x) | Квадратный корень из числа x | sqrt(9) 3 |
| floor(x) | Наибольшее целое, не превосходящее данное число | floor(3.88) 3 |
| round(x) round(x,n) | Округление данного числа x до n знаков после запятой | round(3.456,2) 3.46 |
| ceiling(x) | Округление в большую сторону | ceiling(3.3) 4 |
| trunc(x) | Отсечение дробной части | trunc(-3.4) -3 |
| signif(x,digits=n) | Округляет x до заданного числа значащих цифр | signif(3.479,2) 3.5 |
| factorial(n) | Факториал n!=1⋅2⋅...⋅n и 0!=1 | factorial(4) 24 |
| choose(n,k) | Число сочетаний Сkn | choose(5,2) 10 |
| exp(x) | Экспонента ex | exp(1) 2.718282 |
| log(x) | Натуральный логарифм lnx | log(3) 1.098612 |
| log10(x) | Десятичный логарифм lgx | log10(1000) 3 |
| log(x, base=y) | Логарифм с основанием y logyx | log(8,2) 3 |
| sin(x) | Синус, угол x в радианах | sin(pi) 0 |
| cos(x) | Косинус, угол x в радианах | cos(pi) -1 |
| tan(x) | Тангенс, угол x в радианах | tan(pi) 0 |

В качестве примера работы с числовыми объектами, переменными и выражениями рассмотрим задачу:

Рассчитать месячные выплаты (pl\_mounth) и суммарную выплату (summa\_vipl) по кредиту. О кредите известно, что он составляет 100000 рублей (summa\_cred), берется на 3 года (let), под 15 процентов (pr).

*Решение*

summa\_cred <- 100000 *#сумма кредита*

let <- 3 *#срок кредита, лет*

pr <- 15 *#процентная ставка*

pr <- pr/100 *#относительная величина процентной ставки*

pl\_mounth <- (summa\_cred \* pr \* (1+pr) ^ let) / (12 \* ((1 + pr) ^ let - 1)) *#месячный платеж*

pl\_mounth *#вывод значения месячного платежа*

summa\_vipl <- (pl\_mounth \* 12) \* let *#сумма выплат за весь срок кредита*

summa\_vipl *#вывод суммы выплат*

*Результаты:*

[1] 3649.808 – месячный платеж

[1] 131393.1 – сумма выплат

## 1.8 Числовые векторы и операции с ними

Вектор – это некоторый список данных одного типа (аналог одномерных массивов). Для создания вектора используется функция ***c( )***, которая связывает элементы в одну структуру под названием вектор. Почему векторы в R важны? Дело в том, что вектор является основной единицей хранения данных. И даже датафрейм воспринимается как набор векторов. Если рассматривать специфические характеристики модели, то они также представлены векторами (например, вектор с техническими характеристиками, вектор с показателями модели качества и др.).

* Создадим вектор v из трех элементов. Элементы вектора записываются через запятую.

v <- c(3, 5, 9)

v

[1] 3, 5, 9

* Создадим вектор с показателями объемов продаж разных фирм sales:

sales <- c(30, 80, 24, 67, 90, 32, 24)

* Выясним, по скольким фирмам у нас есть данные, т.е. найдем длину вектора:

length (sales)

[1] 7

Рассмотрим, как обращаться к элементам вектора, если нам необходимы показатели конкретных фирм, или одной. Обращение к элементам осуществляется по индексам (порядковому номеру элемента в этом векторе). Возможно три стандартных варианта: выбираем один элемент по его индексу, выбираем несколько произвольных индексов, или выбираем несколько последовательно расположенных элементов, зная стартовый индекс и конечный. Индекс элемента указывается в квадратных скобках. Нумерация в R начинается с 1.

* Выбираем первый элемент:

sales [1]

[1] 30

* Если задать индекс равный 0, то R выдаст пустой элемент числового типа (numeric), так как вектор sales содержит числа.

sales [0]

numeric(0)

* Выбираем последний элемент. В R нет инструмента, чтобы отсчитывать элементы с конца, поэтому последний элемент придется выбирать при помощи функции вычисления длины вектора:

sales [length(sales)]

[1] 24

* Для выбора нескольких произвольных элементов, необходимо оформить вектор. Выбрать элементы 1, 3, 6:

sales [c(1,3,6)]

[1] 30 24 32

* Для выбора последовательных элементов, необходимо использовать последовательность из целых чисел. Выберем элементы с 1 по 3 включительно.

sales[1:3]

[1] 30 80 24

* Для изменения значения элемента вектора, необходимо указать индекс и присвоить новое значение. Первый элемент изменим на 44.

sales [1] <- 44

sales

[1] 44 80 24 67 90 32 24

* Для замены нескольких элементов необходимо указать срез замены и присвоить вектор с новыми значениями

sales [2:3] <- c(56, 88)

sales

[1] 44 56 88 67 90 32 24

* Для выбора элементов по условию необходимо в квадратных скобках ввести условие:

sales[sales > 30]

[1] 44 56 88 67 90 32 24

* Для вывода фирмы с конкретным показателем продаж, необходимо использовать оператор which():

which (sales == 24)

[1] 24

which (sales > 30)

[1] 1 2 3 4 5 6

* Вектора очень удобны на практике. Возвести все элементы в квадрат:

sales \*\* 2

* Сложить два вектора одинаковой длины

v1 <- c(0, 1, 4)

v2 <- c(8, 9, 2)

v1 + v2

[1] 8 10 6

v1 – v2

[1] -8 -8 2

* Округлить значения элементов вектора

round (c(3.4, 5.7, 0.1))

[1] 3 6 0

* Найти сумму всех элементов

sum(v2)

[1] 19

* Найти произведение всех элементов

prod(v2)

[1] 144

* Сортировка элементов по возрастанию

sort (v2)

[1] 2 8 9

* Сортировка элементов по убыванию

sort (v2, descreasing = TRUE)

[1] 9 8 2

# 2. Задания для самостоятельной работы

1. Вычислить в R:

* 673 – 1122
* log(125)
* log3(81)

1. В векторе flights\_d сохранено число вылетов из аэропорта А, а в векторе flights\_a — число прилётов в этот аэропорт за неделю.

flights\_d <- c(140, 150, 100, 90, 230, 240, 165)  
flights\_a <- c(65, 145, 80, 87, 220, 268, 216)

Используя R, ответьте на вопросы. Приведите код R, используемый для ответа на вопросы.

2.1. Сколько вылетов из аэропорта А было зафиксировано в среду?

2.2. На сколько число вылетов во вторник больше числа прилётов во вторник?

2.4. Сколько всего вылетов из аэропорта А было зафиксировано за неделю?

2.5. В какие дни недели число прилётов не превышает 220? Ответ подразумевает номера дней недели от 1 до 7.

1. В векторе pos сохранены значения весов посылок в килограммах:

pos <- c(4.765, 3.230, 1.256, 1.780, 2.583, 2.781, 3.945, 2.345)

3.1. Используя R, выведите ответы на вопросы.

* Сколько всего посылок было взвешено?
* Какой вес был у самой тяжёлой посылки? А у лёгкой?

3.2. Создайте вектор pos.round со значениями весов посылок в килограммах, округлённых в меньшую сторону.

3.3. Создайте вектор pos\_g со значением весов посылок в граммах.

1. Согласно данным сайта <https://milknews.ru/> на сентябрь 2019 года, средние цены за килограмм сырого молока в Ненецком автономном округе, Камчатском крае, Москве, Республике Бурятия, Ивановской и Ленинградской областях были следующими:

milk <- c(89.5, 50.5, 31.5, 21.0, 22.1, 27.4)

4.1. Не используя готовых функций для вычисления среднего значения, посчитайте среднее арифметическое по вектору milk и сохраните его в переменную (назовите её по своему усмотрению).

4.2. Одним из показателей разброса значений относительно среднего является выборочная дисперсия, которая вычисляется следующим образом:

где – номер элемента в векторе, – среднее арифметическое, – число элементов в векторе. Другими словами, для вычисления дисперсии необходимо из каждого элемента вычесть среднее, возвести полученные разности в квадрат, все просуммировать и поделить на число элементов, уменьшенное на 1.

Используя результат пункта 4.1, вычислите выборочную дисперсию для вектора milk. Сохраните результат в переменную (назовите её по своему усмотрению) и выведите на экран сообщение в одну строку вида «выборочная дисперсия равна: значение», где вместо слова «значение» подставлен полученный результат.

1. Встроенный в R вектор euro содержит стоимость 1 евро в национальных европейских валютах на 1998 год:

euro

## ATS BEF DEM ESP FIM FRF   
## 13.760300 40.339900 1.955830 166.386000 5.945730 6.559570   
## IEP ITL LUF NLG PTE   
## 0.787564 1936.270000 40.339900 2.203710 200.482000

Например, число 166.386 под ESP означает, что 1 евро равняется 166.386 испанским песетам.

5.1. Запросите помощь по вектору euro и прочитайте информацию про него.

5.2. Используя элементы вектора euro, определите, какой сумме в евро соответствует

* 100 финских марок;
* 50 бельгийских франков.

5.3. Используя элементы вектора euro, определите, в какой европейской валюте 1 евро «весит» больше всего. Ответ необходимо дать в виде индекса этой валюты в векторе euro. Решение может занимать более одной строки. Использовать готовые функции, не обсуждаемые в рамках практической работы, например готовую функцию max(), нельзя.

6. Известно, что в таблице хранятся показатели по трём странам за пять лет. Фрагмент таблицы выглядит так:

|  |  |
| --- | --- |
| **country** | **year** |
| France | 2000 |
| France | 2001 |
| France | 2002 |
| France | 2003 |
| France | 2004 |
| Italy | 2000 |
| Italy | 2001 |
| Italy | 2002 |
| Italy | 2003 |
| Italy | 2004 |
| Spain | 2000 |
| Spain | 2001 |
| Spain | 2002 |
| Spain | 2003 |
| Spain | 2004 |

6.1. Создайте вектор country с названиями стран, то есть вектор, который послужил бы первым столбцом таблицы выше.

6.2. Создайте вектор year с годами, который мог бы послужить вторым столбцом таблицы выше.

*Подсказка.* Векторы, содержащие повторяющиеся значения, можно создавать с помощью функции *rep()*, от английского repeat.