

Оглавление

Отзывы о книге.....	16
Благодарности.....	18
Введение	20
Зачем использовать R в статистических вычислениях?	20
Объектно-ориентированное программирование	22
Функциональное программирование	23
Для кого написана эта книга?.....	23
Немного о себе.....	25
От издательства	25
Глава 1. Первые шаги	26
1.1. Как запустить R	26
1.1.1. Интерактивный режим	26
1.1.2. Пакетный режим.....	28
1.2. Первый сеанс R.....	29
1.3. Знакомство с функциями	32
1.3.1. Область видимости переменной.....	34
1.3.2. Аргументы по умолчанию	35
1.4. Важнейшие структуры данных R	36
1.4.1. Векторы	36
1.4.2. Символьные строки.....	37
1.4.3. Матрицы	38
1.4.4. Списки	39
1.4.5. Кадры данных.....	41
1.4.6. Классы.....	41
1.5. Расширенный пример: регрессионный анализ экзаменационных оценок.....	42
1.6. Запуск и завершение	46

1.7. Получение справочной информации	47
1.7.1. Функция help()	48
1.7.2. Функция example()	48
1.7.3. Если вы в общих чертах представляете, что ищете	50
1.7.4. Справка по другим темам	50
1.7.5. Справка по пакетному режиму	51
1.7.6. Справочная информация в интернете	51
Глава 2. Векторы	53
2.1. Скаляры, векторы, массивы и матрицы	53
2.1.1. Добавление и удаление элементов векторов	54
2.1.2. Получение длины вектора	55
2.1.3. Матрицы и массивы как векторы.....	56
2.2. Объявления	56
2.3. Переработка	58
2.4. Основные операции с векторами	59
2.4.1. Арифметические и логические операции с векторами.....	59
2.4.2. Индексирование векторов.....	60
2.4.3. Генерирование векторов оператором :	61
2.4.4. Генерирование векторных последовательностей функцией seq().....	62
2.4.5. Повторение векторных констант функцией rep().....	63
2.5. all() и any()	63
2.5.1. Расширенный пример: поиск серий последовательных единиц	64
2.5.2. Расширенный пример: прогнозирование временных рядов с дискретными значениями	66
2.6. Векторизованные операции	69
2.6.1. Вектор на входе, вектор на выходе.....	69
2.6.2. Вектор на входе, матрица на выходе	72
2.7. NA и NULL	73
2.7.1. Значение NA	73
2.7.2. Значение NULL	74
2.8. Фильтрация	75
2.8.1. Генерирование индексов фильтрации.....	75
2.8.2. Фильтрация с использованием функции subset()	77
2.8.3. Функция выбора which()	77
2.9. Векторизованная конструкция if-then-else: функция ifelse().....	78
2.9.1. Расширенный пример: мера связи	79

2.9.2. Расширенный пример: перекодирование набора данных	82
2.10. Проверка равенства векторов	85
2.11. Имена элементов векторов	87
2.12. Подробнее о <code>c()</code>	88
Глава 3. Матрицы и массивы	89
3.1. Создание матриц	89
3.2. Общие операции с матрицами	91
3.2.1. Выполнение операций линейной алгебры с матрицами.....	91
3.2.2. Индексирование матриц	92
3.2.3. Расширенный пример: обработка графических изображений.....	93
3.2.4. Фильтрация матриц	97
3.2.5. Расширенный пример: генерирование ковариационной матрицы.....	99
3.3. Применение функций к строкам и столбцам матриц.....	101
3.3.1. Использование функции <code>apply()</code>	101
3.3.2. Расширенный пример: поиск выбросов	103
3.4. Добавление и удаление строк и столбцов матриц	105
3.4.1. Изменение размера матрицы	105
3.4.2. Расширенный пример: поиск ближайшей пары вершин в графе.....	107
3.5. Чем векторы отличаются от матриц.....	111
3.6. Предотвращение непреднамеренного снижения размерности.....	112
3.7. Назначение имен для строк и столбцов матриц.....	114
3.8. Массивы более высокой размерности.....	114
Глава 4. Списки.....	117
4.1. Создание списков	117
4.2. Общие операции со списками	119
4.2.1. Индексирование списков.....	119
4.2.2. Добавление и удаление элементов списка	120
4.2.3. Получение размера списка.....	122
4.2.4. Расширенный пример: поиск слов.....	122
4.3. Обращение к компонентам списков и значениям	125
4.4. Применение функций к спискам.....	127
4.4.1. Функции <code>lapply()</code> и <code>sapply()</code>	127
4.4.2. Расширенный пример: поиск слов (продолжение).....	128
4.4.3. Расширенный пример: возвращение к данным абалонов	131
4.5. Рекурсивные списки.....	132

Глава 5. Кадры данных	134
5.1. Создание кадров данных.....	134
5.1.1. Обращение к кадрам данных	135
5.1.2. Расширенный пример: регрессионный анализ экзаменационных оценок, продолжение.....	136
5.2. Другие матричные операции.....	137
5.2.1. Извлечение подкадров данных	137
5.2.2. Об интерпретации значений NA.....	138
5.2.3. Использование rbind() и cbind() и альтернативных функций	139
5.2.4. Применение apply().....	141
5.2.5. Расширенный пример: анализ зарплаты	141
5.3. Слияние кадров данных	143
5.3.1. Расширенный пример: база данных работников.....	145
5.4. Применение функций к кадрам данных.....	146
5.4.1. Функции lapply() и sapply() с кадрами данных	147
5.4.2. Расширенный пример: применение моделей логистической регрессии	147
5.4.3. Расширенный пример: изучение китайских диалектов	149
Глава 6. Факторы и таблицы	157
6.1. Факторы и уровни.....	157
6.2. Типичные функции, используемые с факторами	158
6.2.1. Функция tapply()	159
6.2.2. Функция split()	160
6.2.3. Функция by().....	162
6.3. Работа с таблицами	164
6.3.1. Операции матриц/массивов с таблицами	166
6.3.2. Расширенный пример: извлечение подтаблицы	168
6.3.3. Расширенный пример: поиск максимальных ячеек в таблице.....	171
6.4. Другие функции для работы с факторами и таблицами.....	173
6.4.1. Функция aggregate()	173
6.4.2. Функция cut().....	174
Глава 7. Программные конструкции	175
7.1. Управляющие команды	175
7.1.1. Циклы	175
7.1.2. Перебор не векторных множеств.....	178
7.1.3. if-else	179
7.2. Арифметические и логические операторы и значения.....	181

7.3. Значения по умолчанию для аргументов.....	183
7.4. Возвращаемые значения.....	183
7.4.1. Нужен ли явный вызов <code>return()</code> ?.....	184
7.4.2. Возвращение сложных объектов.....	185
7.5. Функции как объекты.....	186
7.6. Окружение и проблемы видимости.....	188
7.6.1. Окружение верхнего уровня.....	189
7.6.2. Иерархия видимости.....	189
7.6.3. Подробнее о <code>ls()</code>	193
7.6.4. Функции (почти) не имеют побочных эффектов.....	194
7.6.5. Расширенный пример: функция для вывода содержимого кадра вызовов.....	195
7.7. В языке R нет указателей.....	197
7.8. Восходящая запись.....	199
7.8.1. Запись в нелокальные переменные с использованием оператора суперприсваивания.....	199
7.8.2. Запись в нелокальные переменные с использованием <code>assign()</code>	201
7.8.3. Расширенный пример: дискретно-событийное моделирование в R.....	202
7.8.4. Когда следует использовать глобальные переменные?.....	210
7.8.5. Замыкания.....	214
7.9. Рекурсия.....	216
7.9.1. Реализация быстрой сортировки.....	216
7.9.2. Расширенный пример: бинарное дерево поиска.....	217
7.10. Функции замены.....	223
7.10.1. Что считается функцией замены?.....	224
7.10.2. Расширенный пример: класс вектора с хранением служебной информации.....	225
7.11. Средства организации кода функций.....	227
7.11.1. Текстовые редакторы и интегрированные среды разработки.....	227
7.11.2. Функция <code>edit()</code>	228
7.12. Написание собственных бинарных операций.....	229
7.13. Анонимные функции.....	229
Глава 8. Математические вычисления и моделирование в R.....	231
8.1. Математические функции.....	231
8.1.1. Расширенный пример: вычисление вероятности.....	232
8.1.2. Накапливаемые суммы и произведения.....	233

8.1.3. Минимумы и максимумы	233
8.1.4. Численные методы.....	234
8.2. Функции статистических распределений.....	235
8.3. Сортировка	236
8.4. Операции линейной алгебры с векторами и матрицами	238
8.4.1. Расширенный пример: векторное произведение	241
8.4.2. Расширенный пример: нахождение стационарных распределений для цепей Маркова.....	242
8.5. Операции с множествами.....	245
8.6. Имитационное моделирование в R.....	247
8.6.1. Встроенные генераторы случайных величин	247
8.6.2. Получение одной случайной серии при повторных запусках	248
8.6.3. Расширенный пример: комбинаторное моделирование	249
Глава 9. Объектно-ориентированное программирование.....	251
9.1. Классы S3	251
9.1.1. Обобщенные функции S3.....	252
9.1.2. Пример: ООП в функции линейной модели lm()	252
9.1.3. Поиск реализаций обобщенных методов	254
9.1.4. Написание классов S3.....	256
9.1.5. Наследование	258
9.1.6. Расширенный пример: класс для хранения верхних треугольных матриц.....	259
9.1.7. Расширенный пример: полиномиальная регрессия	264
9.2. Классы S4	268
9.2.1. Написание классов S4.....	268
9.2.2. Реализация обобщенной функции в классе S4	270
9.3. S3 и S4	271
9.4. Управление объектами	272
9.4.1. Вывод списка объектов функцией ls()	272
9.4.2. Удаление конкретных объектов функцией rm().....	272
9.4.3. Сохранение коллекции объектов функцией save()	273
9.4.4. «Что это такое?».....	274
9.4.5. Функция exists().....	276
Глава 10. Ввод/вывод	277
10.1. Работа с клавиатурой и монитором	277
10.1.1. Использование функции scan()	277

10.1.2. Функция <code>readline()</code>	280
10.1.3. Вывод на экран	280
10.2. Чтение и запись файлов	281
10.2.1. Чтение кадров данных или матриц из файлов.....	281
10.2.2. Чтение текстовых файлов	283
10.2.3. Соединения.....	283
10.2.4. Расширенный пример: чтение файлов данных PUMS.....	285
10.2.5. Обращение к файлам на удаленных машинах по URL-адресам	289
10.2.6. Запись в файл.....	290
10.2.7. Получение информации о файлах и каталогах	292
10.2.8. Расширенный пример: суммирование содержимого многих файлов	292
10.3. Доступ в интернет.....	293
10.3.1. Обзор TCP/IP.....	294
10.3.2. Сокеты в R	295
10.3.3. Расширенный пример: параллелизм в R.....	296
Глава 11. Работа со строками.....	299
11.1. Обзор функций для работы со строками	299
11.1.1. <code>grep()</code>	299
11.1.2. <code>nchar()</code>	300
11.1.3. <code>paste()</code>	300
11.1.4. <code>sprintf()</code>	300
11.1.5. <code>substr()</code>	301
11.1.6. <code>strsplit()</code>	301
11.1.7. <code>regexpr()</code>	301
11.1.8. <code>gregexpr()</code>	301
11.2. Регулярные выражения.....	302
11.2.1. Расширенный пример: проверка имени файла на наличие определенного суффикса	303
11.2.2. Расширенный пример: формирование имен файлов.....	304
11.3. Применение строковых функций в режиме отладки <code>edtdbg</code>	306
Глава 12. Графика	308
12.1. Построение графиков	308
12.1.1. Основная функция базовой графики R: <code>plot()</code>	308
12.1.2. Рисование линий: функция <code>abline()</code>	309
12.1.3. Создание нового графика при сохранении старых.....	311

12.1.4. Расширенный пример: две оценки плотности на одном графике.....	311
12.1.5. Расширенный пример: подробнее о примере полиномиальной регрессии.....	313
12.1.6. Добавление точек: функция <code>points()</code>	317
12.1.7. Добавление условных обозначений: функция <code>legend()</code>	317
12.1.8. Добавление текста: функция <code>text()</code>	317
12.1.9. Функция <code>locator()</code>	318
12.1.10. Восстановление графика.....	319
12.2. Настройка графиков	320
12.2.1. Изменение размера символов: аргумент <code>sex</code>	320
12.2.2. Изменение диапазонов осей: аргументы <code>xlim</code> и <code>ylim</code>	320
12.2.3. Добавление многоугольника: функция <code>polygon()</code>	323
12.2.4. Сглаживание наборов точек: функции <code>lowess()</code> и <code>loess()</code>	324
12.2.5. Построение графиков конкретных функций	324
12.2.6. Расширенный пример: увеличение части кривой	325
12.3. Сохранение графиков в файлах	328
12.3.1. Графические устройства R	328
12.3.2. Сохранение выведенного графика	329
12.3.3. Закрытие графического устройства R.....	329
12.4. Создание трехмерных графиков.....	330
Глава 13. Отладка.....	332
13.1. Фундаментальные принципы отладки	332
13.1.1. Суть отладки: принцип подтверждения.....	332
13.1.2. Запуск <code>Small</code>	333
13.1.3. Модульная нисходящая отладка.....	333
13.1.4. Защитное программирование.....	334
13.2. Для чего использовать отладочные средства?	334
13.3. Использование отладочных средств R.....	335
13.3.1. Пошаговое выполнение кода функциями <code>debug()</code> и <code>browser()</code>	335
13.3.2. Использование команд просмотра.....	335
13.3.3. Назначение точек прерывания.....	336
13.3.4. Функция <code>trace()</code>	338
13.3.5. Выполнение проверок после сбоя функциями <code>traceback()</code> и <code>debugger()</code>	338
13.3.6. Расширенный пример: два полных сеанса отладки	339
13.4. На пути прогресса: более удобные средства отладки.....	348

13.5. Обеспечение согласованности в отладочном коде	351
13.6. Синтаксические ошибки и ошибки времени выполнения	351
13.7. Применение GDB с кодом R	352
Глава 14. Улучшение быстродействия: скорость и память.....	354
14.1. Написание быстрого кода R	354
14.2. Ужасающий цикл for	355
14.2.1. Векторизация и ускорение выполнения кода.....	355
14.2.2. Расширенный пример: ускорение моделирования методом Монте-Карло	357
14.2.3. Расширенный пример: генерирование матрицы степеней	361
14.3. Функциональное программирование и работа с памятью	363
14.3.1. Присваивание векторов	363
14.3.2. Копирование при изменении	364
14.3.3. Расширенный пример: предотвращение копирования в памяти	365
14.4. Использование Rprof() для поиска мест замедления в коде.....	366
14.4.1. Мониторинг с использованием Rprof().....	366
14.4.2. Как работает Rprof()	368
14.5. Компиляция в байт-код	370
14.6. О нет, данные не помещаются в памяти!	370
14.6.1. Создание блоков	371
14.6.2. Применение пакетов R для управления памятью	371
Глава 15. Взаимодействие R с другими языками	373
15.1. Написание функций C/C++ для вызова из R	373
15.1.1. Что нужно знать для взаимодействия R с C/C++	374
15.1.2. Пример: извлечение поддиагоналей квадратной матрицы	374
15.1.3. Компиляция и запуск кода	375
15.1.4. Отладка кода R/C.....	376
15.1.5. Расширенный пример: прогнозирование временных рядов с дискретными значениями	378
15.2. Использование R из Python	380
15.2.1. Установка RPy	380
15.2.2. Синтаксис RPy.....	381
Глава 16. Параллелизм в R	384
16.1. Проблема взаимных исходящих связей	384
16.2. Пакет snow	385

16.2.1. Выполнение кода snow	386
16.2.2. Анализ кода snow	388
16.2.3. Какого ускорения можно добиться?.....	389
16.2.4. Расширенный пример: кластеризация методом k-средних	389
16.3. Переход на уровень C	392
16.3.1. Использование многоядерных машин.....	393
16.3.2. Расширенный пример: задача взаимных исходящих связей в OpenMP	393
16.3.3. Выполнение кода OpenMP	394
16.3.4. Анализ кода OpenMP	395
16.3.5. Другие директивы OpenMP.....	397
16.3.6. Программирование графических процессоров.....	398
16.4. Общие факторы быстроедействия.....	399
16.4.1. Источники непроизводительных затрат.....	399
16.4.2. Тривиальная параллелизуемость	401
16.4.3. Статическое и динамическое распределение задач.....	403
16.4.4. Программная алхимия: преобразование общих задач в тривиально параллельные	405
16.5. Отладка параллельного кода R	406

Приложение А

Установка R.....	408
А.1. Загрузка R из CRAN.....	408
А.2. Установка из менеджера пакетов Linux.....	408
А.3. Установка из исходного кода	409

Приложение Б.

Установка и использование пакетов	410
Б.1. Основы работы с пакетами	410
Б.2. Загрузка пакета с жесткого диска	410
Б.3. Загрузка пакета из интернета	411
Б.3.1. Автоматическая установка пакетов.....	411
Б.3.2. Ручная установка пакетов.....	412
Б.4. Вывод списка функций в пакете	413