

Осваиваем язык Julia

Совершенствование мастерства в области аналитики и программирования при помощи Julia в целях решения задач комплексной обработки данных

open so

Малколм Шеррингтон

Осваиваем язык Julia

Julia — это хорошо структурированный язык программирования с большим быстродействием, устраняющий классическую проблему выполнения анализа на одном языке и трансляции его результатов на второй с целью повышения производительности. Эта книга поможет вам развить и усовершенствовать свои навыки программирования на Julia для решения задач автоматизации, возникающих в реальной жизни.

Книга начинается с небольшого инструктажа по поводу инсталляции и выполнения Julia в разных операционных средах. Затем, вы сравните самые разные способы работы с языком и подробно изучите его ключевой функционал, разбирая практические примеры, построенные на основе пошагового принципа. Пользуясь простыми статистическими и аналитическими показателями, вы откроете для себя быстродействие языка, его реальную мощь, которая делает его особенно полезным в высокоинтенсивных вычислительных задачах, и отметите, что язык Julia способен сотрудничать с внешними процессами, получая значительное улучшение качества графики и визуализации данных. Наконец, вы займетесь метапрограммированием и узнаете, как оно укрепляет мощь языка и формируют его сетевую и распределенную вычислительную среду.

Для кого эта книга написана

Это практическое руководство предназначено для специалистов в области науки о данных. Книга предполагает наличие некоторых навыков работы с Julia и навыков программирования на скриптовом языке, таком как Python и R, либо на компилируемом языке, таком как C или Java.

Чему вы научитесь, прочитав эту книгу

- Инсталлировать среду разработки и выполнять ее сборку и настройку под вашу операционную среду
- Создавать проекты в области науки о данных в рамках полного цикла ETL, анализа и визуализации данных
- Понимать систему типов и принципы множественной диспетчеризации для получения большей отдачи от программирования на Julia
- Взаимодействовать с файлами и таблицами данных с целью изучения простых статистических и аналитических показателей
- Отображать графики и визуальные данные с целью проведения на Julia имитационного моделирования
- Использовать Julia для взаимодействия с базами данных SQL и NoSQL
- Работать с распределенными системами в Веб и в облаке
- Разрабатывать свои собственные программные пакеты и участвовать в сообществе программистов на Julia в качестве соавтора

Об авторе

Малкольм Шеррингтон работает в сфере информационных технологий более 35 лет. Он имеет степени в области математики, химии и инженерно-технических наук и читал лекции в двух различных университетах в Великобритании, а также работал в аэрокосмической и медицинской отраслях экономики. В настоящее время он руководит собственной компанией в финансовом секторе, с определенными интересами к высокоэффективным вычислениям и приложениям на основе графического процессора и параллельных вычислений.

Будучи деятельным специалистом, Malcolm начал программировать научные задачи на Fortran и C, совершенствуя навыки программирования на Ada и Common LISP, и недавно занялся обработкой данных и аналитикой на Perl, Python и R.

Малкольм является организатором Лондонской ассоциации пользователей Julia. Кроме того, он является со-организатором британской meetup-группы по высокопроизводительным вычислениям и финансовым технологиям и Лондонской meetup-группы финансовых аналитиков.

Я бы хотел посвятить эту книгу памяти моей покойной жены, Хэйзел Шеррингтон, без чьего поощрения и поддержки моя причастность к Julia не состоялась бы, но которой уже больше нет здесь, чтобы увидеть кульминацию своего предвидения.

Кроме того, хочу выразить особую благодарность Барбаре Дорэ и Джэймсу Уэймсу за их существенную помощь и материальную поддержку при подготовке этой книги.

О рецензентах

Гурурагав Гопал в настоящее время работает консультантом по управлению рисками в стартапах. Ранее, он работал в Paterson Securities в качестве разработчика в области финансовой аналитики/трейдерского консультанта. Кроме того, работал консультантом по анализу данных и был связан с организацией электронной торговли. Он преподавал студентам и аспирантам в Технологическом университете VIT в Веллуре, Индия, в областях распознавания образов, машинного обучения и больших данных. Как научный сотрудник, связан с несколькими исследовательскими организациями, а именно, IFMR и NAL. Кроме того, выступил рецензентом книги *Изучение интеллектуального анализа данных при помощи R*, Packt Publishing, также рецензентом для нескольких журналов и конференций.

He did his bachelor's degree in electrical and electronics engineering with a master's degree in computer science and engineering. He later did his course work from IFMR in financial engineering and risk management, and since then, he has been associated with the financial industry. He has won many awards and has a few international publications to his credit.

He is interested in programming, teaching, and doing consulting work. During his free time, he listens to music.

He can be contacted for professional consulting through LinkedIn at [in.linkedin.com/in/gururaghavg](https://www.linkedin.com/in/gururaghavg).

Zhuo QL (a.k.a KDr2 online) is a free developer from China who has about 10 years' experience in Linux, C, C++, Java, Python, and Perl development. He loves to participate in and contribute to the open source community (which, of course, includes the Julia community). He maintains a personal website at <http://kdr2.com>; you can find out more about him there.

Dan Wlasiuk is the author of various Julia packages including TimeSeries and Quandl, and he is also the founder of the JuliaQuant GitHub organization of quantitative finance related packages.

Содержание

ПРЕДИСЛОВИЕ	1
О чем эта книга рассказывает	2
Что требуется для этой книги	4
Для кого эта книга	4
Стандарт оформления	4
Отзывы читателей	5
Служба поддержки	5
Скачивание кода примеров	6
Опечатки	6
Пиратство	6
Вопросы	7
Readme от автора	7
Комментарий переводчика	8
Выполнение примеров программного кода на Julia	8
Установка среды разработки Julia	10
Установка пакетов в среду разработки Julia	11
Установка и удаление IDE Juno/Atom	12
Установка и работа с плагином для Eclipse	13
Работа с Julia в облаке JuliaBox	14
Установка и удаление дистрибутива Anaconda Python	14
Запуск сервера записных книжек Jupyter	15
Факультативно	15
1. СРЕДА РАЗРАБОТКИ JULIA	1
Введение	1
Философия	2
Роль в науке о данных и в области больших данных	3
Сопоставление с другими языками	4

Характеристики	7
Начало работы	8
Исходные тексты Julia	8
Сборка из исходников	9
Изучение стэка исходного кода	14
Интегрированная среда разработки Juno	15
Плагин JuliaDT для среды программирования Eclipse	17
Среда программирования IJulia	18
Краткий обзор языка Julia	20
Julia через консоль	21
Установка некоторых пакетов	25
Управление пакетами	28
Вывод перечня пакетов, добавление и удаление	29
Выбор и ознакомление с пакетами	30
Как деинсталлировать Julia	34
Добавление незарегистрированного пакета	35
Особенности языка Julia	35
Параллельная обработка	36
Множественная диспетчеризация	36
Гомоиконные макрокоманды	36
Межязыковое взаимодействие	37
Заключение	37
2. РАЗРАБОТКА НА JULIA	39
Целые числа, биты, байты и логические значения	40
Целое	40
Логические и арифметические операторы	42
Логический тип	43
Массивы	44
Операции на матрицах	46
Поэлементные операции	47
Простая марковская цепь – кошки и мышки	48
Символьный и строковый типы	49
Символьный тип	49
Строковый тип	50
Пример: игра "Быки и коровы"	54
Вещественные, комплексные и рациональные числа	56
Вещественные	57
Рациональные числа	58
Комплексные числа	59
Составные типы	62

Дополнительно о матрицах	63
Векторизованный и девекторизованный программный код	64
Многомерные массивы	65
Разреженные матрицы	67
Массивы и таблицы данных	68
Словари, множества и прочее	70
Словари	70
Множества	72
Другие структуры данных	73
Заключение	74
3. ТИПЫ И ДИСПЕТЧЕРИЗАЦИЯ	75
Функции	75
Объекты первого класса	76
Передача аргументов	80
Область видимости	85
Задача о восьми ферзях	88
Система типов Julia	90
Обзор рационального типа	90
Тип данных для транспортных средств	93
Перечисляемый тип	105
Множественная диспетчеризация	106
Параметрические типы	108
Операции преобразования и приведения	110
Модуль для фиксированного 3D-вектора	112
Заключение	114
4. ФУНКЦИОНАЛЬНАЯ СОВМЕСТИМОСТЬ	116
Взаимодействие с другими программными средами	116
Обращение к программам на C и Fortran	117
Язык Python	123
Некоторые другие языки	125
Программный интерфейс Julia	126
Обращение к API из C	127
Метапрограммирование	129
Символические имена	130
Макрокоманды	131
Тестирование	135
Макрокоманда <code>enum</code>	137
Объект <code>Task</code> и многозадачность	140

Параллельные операции	141
Распределенные массивы	144
Простая модель MapReduce	147
Исполнение команд	148
Запуск команд на исполнение	149
Работа с файловой системой	152
Перенаправление ввода-вывода и конвейеры	154
Заключение	157
5. РАБОТА С ДАННЫМИ	159
Базовая система ввода-вывода	159
Терминальный ввод-вывод	159
Дисковые файлы	162
Обработка текста	164
Двоичные файлы	167
Структурированные наборы данных	169
Файлы CSV и DLM	169
Файлы HDF5	174
Файлы XML	178
DataFrames и RDatasets	180
Пакет DataFrames	180
Таблицы данных DataFrame	182
Пакет RDatasets	185
Статистика	191
Простые статистики	192
Выборки и их оценка	194
Модуль Pandas	195
Отдельные темы	196
Временные ряды	196
Распределения вероятностей	200
Проверка статистических гипотез	203
Обобщенная линейная модель	205
Заключение	208
6. НАУЧНОЕ ПРОГРАММИРОВАНИЕ	210
Линейная алгебра	211
Система уравнений	212
Разложение матриц	214
Собственные значения и собственные векторы	215
Матрицы специального вида	218

Обработка сигналов	219
Частотный анализ	220
Фильтрация и сглаживание	220
Цифровые фильтры	223
Обработка изображений	224
Дифференциальные уравнения	226
Решение обыкновенных дифференциальных уравнений	226
Нелинейные обыкновенные дифференциальные уравнения	229
Дифференциальные уравнения в частных производных	233
Оптимизационные задачи	235
Пакет JuMP	236
Пакет Optim	238
Пакет NLOpt	242
Стохастические задачи	245
Стохастическое моделирование	246
Байесовские методы и марковские процессы	251
Заключение	257
7. ГРАФИКА	259
Базовая графика в Julia	260
Построение текстовых графиков	260
Пакет Cairo	262
Пакет Winston	264
Визуализация данных	268
Пакет Gadfly	269
Пакет Compose	274
Графические движки	277
Пакет PyPlot	277
Пакет Gaston	281
Пакет PGFPlots	285
Использование Веб	287
Пакет Vokeh	287
Пакет Plotly	289
Растровая графика	293
Пакет Cairo (повторно)	294
Пакет Winston (повторно)	295
Пакеты Images и ImageView	296
Заключение	299
8. БАЗЫ ДАННЫХ	300

Общий обзор баз данных	300
Вот две таблетки: красная и синяя. Выбирай!	301
Взаимодействие с базами данных	302
Другие соображения	304
Реляционные базы данных	305
Создание и загрузка	306
Нативные интерфейсы	310
Программный интерфейс ODBC	313
Другие методы взаимодействия	318
Интерфейс DBI	319
Пакет PyCall	322
Стандарт взаимодействия JDBC	324
Хранилища данных NoSQL	325
Системы ключ-значение	326
Документно-ориентированные хранилища данных	330
Взаимодействие с RESTful	334
Данные в формате JSON	335
Интернет-СУБД	337
Графовые системы хранения данных	340
Заключение	343
9. СЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ	345
Сокеты и серверы	345
Стандартные порты	346
Сокеты UDP и TCP в Julia	347
"Зазеркальный" эхо-сервер	348
Именованные каналы	352
Работа с Веб	352
Веб-служба на основе TCP	353
Группа пакетов JuliaWeb	355
Сервер цитат	358
Технология WebSocket	361
Обмен сообщениями	366
Электронная почта	366
Социальная сеть Twitter	368
СМС-сообщения	370
Облачные службы	373
Введение в веб-службы Amazon	374
Пакет AWS.jl	376
Платформа Google Cloud	381
Заключение	384

10. РАБОТА С JULIA	386
Внутреннее устройство	386
Язык Fmtolisp	387
Программный интерфейс Julia	389
Генерация машинных кодов	391
Советы относительно производительности	394
Наиболее успешная практика	395
Профилирование	397
Статический анализ кода	399
Отладка	402
Разработка пакета	404
Анатомия	405
Классификация	408
Использование Git	410
Публикация	412
Сообщества программистов	413
Классификации	414
Группа пакетов JuliaAstro	415
Группа пакетов JuliaGPU	422
Что не вошло в книгу?	426
Заключение	427
ПРИЛОЖЕНИЕ	430
Символическая математика с Julia	430
Введение	430
О пакете	431
Уравнения	432
Элементарная алгебра	433
Решение уравнений	433
Построение графиков выражений	437
Пределы	438
Производные	440
Экстремумы	441
Интегрирование	442
Применения	443
ГЛОССАРИЙ ОСНОВНЫХ ТЕРМИНОВ И СОКРАЩЕНИЙ	446
ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ	456

Предисловие

Julia — это относительно молодой язык программирования. Первоначальные проектные работы по языку Julia начались в Массачусетском технологическом институте¹ (MIT) в августе 2009, и к февралю 2012 он стал общедоступным. Данный язык является в основном результатом работы трех разработчиков: Стефана Карпински, Джеффа Безансона и Вирала Шаха. Все трое, вместе с Аланом Эдельманом, до сих пор остаются активно преданными проекту Julia в MIT, где в настоящее время проводится ряд курсов по Julia, многие из которых доступны в Интернет.

Первоначально Julia планировался разработчиками как язык для научного программирования с быстродействием, достаточным для восполнения потребности в моделировании на интерактивном языке с последующей неизбежной переработкой кода на компилирующем языке, таком как C или Fortran. В то время главные языки для научного программирования были защищены правами интеллектуальной собственности, как например, MATLAB и Mathematica, и до сих пор остаются относительно медленными. При этом имеются клоны этих языков в области открытого программного обеспечения, как например, GNU Octave и Scilab, но они работают еще медленнее. Когда язык был запущен, сообщество программистов увидело в Julia замену для MATLAB, однако это не совсем так. Несмотря на то, что синтаксис Julia похож на MATLAB, настолько, что любой компетентный в MATLAB, может легко изучить Julia, он не разрабатывался как клон. Это язык с более развитым набором функций и со многими существенными различиями, которые будут подробно рассмотрены далее.

Период с 2009 засвидетельствовал рост двух новых вычислительных дисциплин: больших данных/облачных вычислений и науки о данных. Обработка больших данных на Hadoop традиционно рассматривается как область программирования на Java, поскольку Hadoop выполняется в виртуальной машине Java. Разумеется, существует возможность обрабатывать большие данные при помощи

¹ Массачусетский технологический институт — университет и исследовательский центр, расположенный в Кембридже, штат Массачусетс, США. Также известен как Массачусетский институт технологий и Массачусетский технологический университет.

языков программирования помимо тех, которые основаны на Java и используют парадигму `jar`-файлов с потоковой передачей, и тут Julia может применяться по аналогии с тем, как это делается на C++, C# и Python.

Появление науки о данных возвестило о начале использования языков программирования, которые просты для аналитиков, обладающих некими навыками программирования, но которые не являются по преимуществу программистами. Два языка, чье развитие ускорилося, чтобы заполнить этот пробел, стали R и Python. Оба они относительно стары, уходя своими корнями в 1990-е годы. Однако популярность обоих демонстрирует стремительный рост, по иронии, примерно с того времени, когда общественности был представлен язык Julia. Но даже имея такую признанную и солидную оппозицию, этот язык взволновал научное программистское сообщество и продолжает проводить рейды в этом пространстве.

Цель этой книги состоит в том, чтобы охватить все аспекты языка Julia, которые делают его привлекательным для аналитиков данных. Язык развивается быстро. Двоичные дистрибутивы доступны для Linux, Mac OS X и Windows, но они отстают от текущих исходников. Поэтому, чтобы выполнять при помощи Julia серьезную работу, важно понимать, как получать и собирать рабочую систему из исходных текстов. Кроме того, для Julia доступны интерактивные среды разработки (IDE), и в данной книге будут проанализированы IDE Jupyter, Juno и плагин JuliaDT для среды Eclipse.

О чем эта книга рассказывает

Глава 1, Среда разработки Julia, рассказывает о практических шагах, которые необходимо предпринять для того, чтобы запустить дистрибутив Julia и привести его в состояние готовности к работе. Важно уметь получать самые последние исходные тексты и собирать систему с нуля, а также находить и устанавливать надлежащие пакеты и при необходимости их удалять.

Глава 2, Разработка на Julia, содержит краткий обзор части стандартных синтаксических конструкций языка. Julia — язык новый, но он отнюдь не такой новый для читателей с компетенцией в MATLAB, R или Python, и поэтому цель главы состоит в том, чтобы с помощью примеров кратко довести до читателей информацию о Julia и направить их к онлайн-новым источникам. Кроме того, важно понимать разницу работы через консоль и с интегрированными средами разработки Jupyter/Juno/JuliaDT.

Глава 3, Типы и диспетчеризация, посвящена системе типов Julia и показывает, каким образом она предоставляет разработчику мощные методы посредством ее *de facto* функциональной системы диспетчеризации.

Глава 4, Функциональная совместимость, касается методов, посредством которых Julia может взаимодействовать с операционной системой и другими языками программирования. Эти методы являются в основном нативными для Julia.

Глава заканчивается введением в параллелизм, который будет рассмотрен подробнее в *Главе 9, Сетевое взаимодействие*.

Глава 5, Работа с данными, рассматривает выполняемую аналитиком данных процедуру, начиная с источника данных и заканчивая результатами анализа. Большинство проектов начинается с данных, которые нужно прочесть, очистить и отобразить. Глава начинается с этого и в дальнейшем описывает простые статистические и аналитические показатели.

Глава 6, Научное программирование, как таковое, можно рассматривать в качестве главной причины для программирования на Julia. Мощь языка заключена в его быстродействии в сочетании с простотой разработки на языке сценариев, что делает его особенно ценным при решении задач с процессами, вычислительно ограниченными возможностями ЦПУ. В главе рассматриваются различные подходы, используемые при решении математических и естественно-научных задач.

Глава 7, Графика, в Julia часто не выдерживает сравнения с другими альтернативными языками, такими как MATLAB и R. В то время как более ранние версии языка имели достаточно ограниченную поддержку графики, теперь это уже не тот случай. Данная глава описывает большое разнообразие сложных подходов к визуализации графики на экране и сохранению ее в дисковых файлах.

Глава 8, Базы данных, рассказывает о взаимодействии языка Julia с базами данных. Данные для анализа могут храниться в базе данных либо результаты анализа требуется сохранить в базе данных. В этой главе будут проанализировано несколько подходов к хранению данных в хранилищах SQL и NoSQL. Они не встроены в язык, а скорее полностью опираются на сторонние пакеты, и поэтому в ближайшем будущем могут быть усовершенствованы.

Глава 9, Сетевое взаимодействие, касается аспектов работы с распределенными источниками данных. В науке о данных большие данные и облачные системы становятся все более распространенными, и данная глава рассказывает о сетевом программировании на уровне сокета и взаимодействии через Веб. Кроме того, она посвящена анализу работы Julia в веб-службах Amazon и с вычислительным сервером Google.

Глава 10, Работа с Julia, предназначена для того, чтобы предоставить информацию и побудить продолжить работу, участвуя в совместной разработке на Julia. Это участие может быть в качестве соавтора, вносящего свой вклад в существующий пакет, либо в качестве участника одного из сообществ программистов на Julia.

Что требуется для этой книги

Разработка на Julia может выполняться в любой из известных вычислительных операционных систем: Linux, OS X и Windows. В целях углубленного исследования языка читатель может захотеть получить последние версии и собрать язык из исходников под Linux. Однако для того, чтобы начать работать с языком в любой из трех операционных платформ, предусмотрена простая и удобная установка языка с использованием двоичного дистрибутива. Кроме того, можно отдельно скачать и установить интегрированную среду разработки (IDE) Juno и плагин для среды разработки Eclipse.

Некоторые примеры в более поздних главах, касающиеся поддержки баз данных, сетевого взаимодействия и облачных служб, потребуют установки дополнительных компонентов и использования дополнительных ресурсов, и вопросы их получения будут обсуждаться в соответствующих частях книги.

Для кого эта книга

Эта книга не является введением в программирование, и потому предполагается, что читатель знаком с понятийным аппаратом, по крайней мере, одного языка программирования. Для тех, кто знаком с языками сценариев, такими как Python, R и MATLAB, эта задача не является трудной, как, впрочем, и для тех, кто использует похожие языки, такие как C, Java и C#.

Однако для аналитика данных, возможно с компетенцией в методах аналитики на основе электронных таблиц, таких как Excel, или статистических пакетов, таких как SPSS и Stata, значительная часть текста должна оказаться полезной.

Стандарт оформления

В этой книге вы найдете ряд текстовых стилей, которые выделяют различные виды информации. Вот некоторые примеры этих стилей и объяснение их значения.

Фрагменты программного кода в тексте, имена таблиц баз данных, имена папок, имена файлов, расширения файлов, пути, фиктивные URL, ввод данных пользователем и дескрипторы Twitter показаны следующим образом: "Папка `test` содержит код, который иллюстрирует, как писать тестовые сценарии и применять систему `Base.Test`."

Блок кода выглядит следующим образом:

```
function isAdmin2(_mc::Dict{ASCIIString,UserCreds}, _name::ASCIIString)
    check_admin::Bool = false;
    try
        check_admin = _mc[_name].admin
```

```
catch
    check_admin = false
finally
    return check_admin
end
end
```

А ввод или вывод командной строки записывается следующим образом:

```
julia> include("asian.jl")
julia> run_asian()
```

Новые термины и **важные слова** показаны полужирным шрифтом. Слова, которые вы видите на экране, например, в меню или диалоговых окнах, выглядят в тексте следующим образом: "Однако случаются и другие, такие как в случае перенаправления и ошибки, одно из них печально знаменитое **404, Страница не найдена.**"

- Предупреждения или важные примечания появляются в этом поле.
- Подсказки и приемы появляются тут.
- Дополнения к тексту оригинала книги.

Отзывы читателей

Отзывы наших читателей всегда приветствуются. Сообщите нам, что вы думаете об этой книге — что вам понравилось, а что нет. Обратная связь с читателями для нас очень важна, поскольку она помогает нам формировать названия книг, из которых вы действительно получите максимум полезного.

Отзыв по общим вопросам можно отправить по адресу feedback@packtpub.com, упомянув заголовок книги в теме вашего электронного сообщения.

Если речь о теме, в которой у вас есть экспертные знания, и вы интересуетесь написанием либо содействием в написании книги, то обратитесь к нашему перечню авторов по адресу www.packtpub.com/authors.

Служба поддержки

Теперь, когда вы являетесь гордым владельцем книги издательства Packt, мы предложим вам ряд возможностей с целью помочь вам получать максимум от своей покупки.

Скачивание кода примеров

Вы можете скачать файлы с кодом примеров из вашего аккаунта по адресу <http://www.packtpub.com/> для всех книг издательства Packt Publishing, которые вы приобрели. Если вы купили эту книгу в другом месте, то можно посетить <http://www.packtpub.com/support> и зарегистрироваться там, чтобы получить файлы прямо по электронной почте.

Опечатки

Несмотря на то, что мы приняли все меры, чтобы гарантировать корректность нашего информационного материала, ошибки действительно случаются. Если вы найдете ошибку в одной из наших книг, возможно, в тексте или коде, то мы будем благодарны, если вы сообщите об этом нам. Поступая так, вы способны уберечь других читателей от расстройств и помочь нам улучшать последующие версии этой книги. Если вы найдете какие-либо опечатки, пожалуйста, сообщите о них, посетив страницу <http://www.packtpub.com/submit-errata>, выбрав вашу книгу, нажав на ссылку вызова формы для отправки опечатки **Errata Submission Form** и введя ваши данные об опечатках. Как только ваши данные будут проверены, сообщение будет принято, и данные об опечатке будут загружены на наш веб-сайт или добавлены к любому списку из существующих списков опечаток в разделе Errata под соответствующим заголовком.

Чтобы просмотреть ранее предоставленные опечатки, перейдите на страницу <https://www.packtpub.com/books/content/support> и в поле поиска введите название книги. Запрошенная информация появится под разделом **Errata**.

Пиратство

Пиратство защищенного авторским правом материала в Интернете является хронической проблемой во всех средствах массовой информации. В издательстве Packt мы очень серьезно относимся к защите нашего авторского права и лицензий. Если вы сталкиваетесь с какими-либо недопустимыми копиями наших работ в какой-либо форме в Интернете, пожалуйста, просим вас незамедлительно предоставить нам адрес размещения или название веб-сайта, чтобы мы могли добиваться правовой защиты.

Пожалуйста, свяжитесь с нами по электронному адресу copyright@packtpub.com с ссылкой на предполагаемый пиратский материал.

Мы ценим вашу помощь в защите наших авторов и нашей способности предоставлять вам ценный информационный материал.

Вопросы

Если у вас есть вопрос по каким-либо аспектам этой книги, то вы можете связаться с нами по электронному адресу questions@packtpub.com, и мы приложим все усилия, чтобы решить ваш вопрос.

Readme от автора

По ходу изложения я неоднократно отмечал, что Julia находится в состоянии разработки и даже еще не достиг уровня версии v.1.0.

Одна из целей написания данной книги состояла в том, чтобы помочь вам почувствовать себя комфортно во время изучения среды разработки на Julia и при взаимодействии с сообществом программистов на Julia.

Хотя при выполнении перевода книги приводимый в книге программный код работал в версии v0.4 (и был адаптирован к последней версии v0.4.5) на разных платформах, в следующих версиях могут возникнуть расхождения.

Проблемы могут быть вызваны:

1. Не рекомендованными для использования (устаревшими) функциями или синтаксическими конструкциями. Например, к последней версии v0.4.x был пересмотрен синтаксис для массивов `Any[]`, кортежей, функции `convert()` и пр.
2. Включение пакетов в Базу (стандартную библиотеку)

Например, функции `DateTime` были включены в состав Базы для версии v0.4. Распределенные массивы были изъяты из Базы и существуют в виде пакета `DistributedArrays.jl`, который необходимо включать в среду в версиях v0.4 и выше.

3. Изменения в пакетах

Разработчикам пакетов необходимо некоторое время, чтобы принять к сведению изменения в функционале Базы и модифицировать свой программный код соответствующим образом. В некоторых случаях результатом таких изменений может стать изменение поведения всего пакета.

Кроме того, пакеты могут быть заменены на другие, программный код перераспределен либо пакет реструктурирован, в том числе путем создания подмодулей, изъятия либо переименования функций, модификации структуры типов.

Поэтому:

Если имеются проблемы с программным кодом: загляните в раздел `issues` стан-

дартной библиотеки или конкретного пакета на Github, либо попробуйте погуглить в отношении своей проблемы или задайте вопрос/заявите о проблеме в разделе issues на Github разработчика или на сайте вопросов и ответов для программистов Stackoverflow

(<http://stackoverflow.com/questions/tagged/julia-lang?page=1&sort=newest&pagesize=15>).

Скорее всего вы окажетесь не первый, кто с этим столкнулся.

Комментарий переводчика

На сегодняшний день книга "*Осваиваем язык Julia*" представляет собой наиболее полное описание возможностей программирования на Julia. Год с момента издания оригинала книги, посвященной такому молодому и изменяющемуся языку как Julia — это очень большой срок. Поэтому сразу стоит отметить, что весь материал книги приведен в соответствие с последней версией языка (v0.4.5 на июнь 2016; в оригинале рассматривалась стабильная на тот момент версия 0.3.x) и дополнен свежей информацией.

Книга содержит много аббревиатур и технических терминов из разных областей науки. Для удобства большинство аббревиатур кратко определены в сносках, а для некоторых терминов в силу отсутствия единой терминологии приведены соответствующие варианты наименований или пояснения.

В целом в книге принят сбалансированный подход к теории и программированию. Разумеется, можно найти более подробное изложение аспектов науки о данных и на основе других языков программирования. Однако основная задача книги заключается в том, чтобы познакомить, каким образом можно использовать мощь языка Julia в научных вычислениях, в том числе, используя методы науки о данных.

Книга может быть интересной широкому кругу специалистов, в том числе в области машинного обучения, начинающим аналитикам данных, преподавателям, студентам, а также всем, кто интересуется программированием.

Выполнение примеров программного кода на Julia

Приведенные в книге примеры протестированы в операционных системах Windows 8.1/10 и Linux/Lubuntu 16.4.

Прилагаемые к книге адаптированные и скорректированные примеры программного кода должны находиться в подпапке `julia_projects` домашней папки пользователя (`/home/julia_projects` или `C:\Users\[ИМЯ_ПОЛЬЗОВАТЕЛЯ]\julia_projects`). Ниже приведена структура папки с проектами:

Alice	данные для примеров из папки code и записных книжках Jupyter
code	консольные программы
data	данные для примеров из папки code и записных книжках Jupyter
gaston	скорректированный графический пакет Gaston
images	файлы изображений из книги и полученные в результате выполнения программ
space	материалы для главы 10 (JuliaAstro)
tempo	временные файлы
Глава 01..Глава 10	записные книжки (практически полностью дублируют папку code)
Полезная информация	дополнительные примеры кода, записные книжки и пр.

Рекомендуем всегда начинать консольную сессию с команды:

```
julia> cd(joinpath(homedir(), "julia_projects"))
```

Или аналогичной ей, чтобы сделать папку с проектами Julia текущей. Впрочем эту команду можно прописать в файле `.juliarc.jl`, который находится в папке `homedir()` (`C:\Users\[ИМЯ_ПОЛЬЗОВАТЕЛЯ]` в Windows; образец файла имеется в папке с прилагаемыми примерами), и тогда эта команда будет исполняться при каждом открытии консоли интерпретатора REPL.

В простейшем случае работу большинства приводимых в книге примеров можно проверить в окне консоли интерпретатора REPL, просто копируя в него фрагменты программного кода из буфера обмена (вставка правой кнопкой мыши возле подсказки `julia>`). В Windows вместо консоли Julia можно использовать консоль, например, Cmder (<http://cmder.net/>) с расширенными возможностями, включая эмуляцию команд Unix; она идет в минимальной и полной версиях и не требует установки (Julia запускается изнутри консоли).

Как вариант, можно перенести все примеры в корневую папку пользователя, скажем, в Windows ее расположение будет таким: `C:\Users\[ИМЯ_ПОЛЬЗОВАТЕЛЯ]\julia_projects`. В этом случае для выполнения примеров сначала нужно перейти в эту папку: `;cd julia_projects` и затем оттуда включать примеры в среду Julia для исполнения, используя команду `include`, к примеру, `include("code/asian_option.jl")`.

Остальные подразумевают использование интегрированной среды разработки Juno/Atom, плагина JuliaDT для интегрированной среды разработки Eclipse, локального сервера записных книжек Jupyter и перспективной облачной веб-службы JuliaBox (<https://www.juliabox.org/>), куда можно загружать локальные записные книжки и файлы с программным кодом на Julia для просмотра и исполнения.

В режиме разработки целесообразно в строке запуска консоли Julia прописать следующие переключатели:

```
-q, --quiet — "тихий" запуск, то есть без стартового заголовка
--depwarn={yes|no|error} — активировать/деактивировать депрекационные
предупреждения (error делает их ошибками)
--precompiled={yes|no} — использовать прекомпилированный код из систем-
ного образа, если есть
```

В следующей версии (0.5.0) предполагается новый переключатель:

```
--compilecache={yes|no} — активировать/деактивировать обновляющую
(incremental) прекомпиляцию модулей.
```

В целом строка должна иметь следующий вид:

```
julia.exe -q --precompiled=yes --depwarn=no
```

Установка среды разработки Julia

В Windows надо просто скачать установщик, соответствующий разрядности вашей операционной системы, и установить (<http://julialang.org/downloads/>). В Linux/Ubuntu сначала нужно выполнить небольшую подготовительную работу. Через терминал следует установить компиляторы:

```
sudo apt-get install gcc
sudo apt-get install g++
sudo apt-get install gfortran
```

Компоновщик:

```
sudo apt-get install cmake
```

И Git-клиент (факультативно):

```
sudo apt-get install git
```

И затем приступить собственно к установке Julia. Для операционных систем Ubuntu предусмотрен *Персональный архив пакетов (PPA)*, который предоставляет возможность автоматического обновления программного обеспечения до самой свежей стабильной версии, в данном случае языка Julia. Чтобы воспользоваться PPA и установить Julia в Ubuntu версий 12.04 и выше, в терминале следует выполнить следующие команды:

```
sudo add-apt-repository ppa:staticfloat/juliareleases
sudo add-apt-repository ppa:staticfloat/julia-deps
sudo apt-get update
```

```
sudo apt-get install julia
```

С другой стороны, можно установить ночные сборки, выполнив следующие команды:

```
sudo apt-add-repository ppa:staticfloat/julianightlies
sudo apt-add-repository ppa:staticfloat/julia-deps
sudo apt-get update
sudo apt-get install julia
```

Сборка новых версий происходит каждую ночь. Если вы уже установили Julia и желаете обновиться до последней версии, выполните следующее:

```
sudo apt-get update
sudo apt-get upgrade
```

Установка пакетов в среду разработки Julia

После установки Julia следует выполнить обновление пакетов:

```
Pkg.update()
```

И затем установить ряд пакетов, которые будут использоваться в примерах:

```
Pkg.add("ASCIIPlots")
```

Кроме упомянутого пакета `ASCIIPlots`, аналогичным образом следует установить следующие пакеты: `IJulia`, `Atom`, `Winston`, `Gadfly`, `PyPlot`, `HDF5`, `Match` и некоторые другие. Как правило, при установке пакетов автоматически устанавливаются другие необходимые пакеты, например, `DataStructures`, `DataFrames`, `DataArrays`, `Distributions`, `StatsBase`, `StatsFuns`, `Compose` и др.

Если по какой-то причине в работе пакета возникнет сбой, то, как правило, его повторная сборка решает проблему, как например:

```
Pkg.build("ZMQ")
```

Если пакет выдает депрекационное предупреждение о нецелесообразности использования какой-либо конструкции или выдает ошибку, то ситуацию можно исправить, внося изменения в сам пакет и собрав его заново (кстати, именно так и работает Github). Например, пакет `Match` выдает предупреждение, что вместо модификатора `String` следует использовать `AbstractString`. Чтобы исправить ситуацию, смотрим на предупреждение, там будет указан конкретный файл модуля. Далее открываем папку (на примере Windows) `C:\Users\[ИМЯ_ПОЛЬЗОВАТЕЛЯ]\.julia\v0.4` и находим там папку пакета `Match` и в нем искомый файл.

В данном случае предупреждение было вызвано в `matchmacro.jl` в нижеприведенном блоке (`String` уже заменено на `AbstractString`)

```
# Regex strings (r"[a-z]*")
elseif isexpr(expr, :macrocall) && expr.args[1] == symbol("@r_str")
    append!(info.tests, [:(isa($val, AbstractString)],
        :(Match.ismatch($expr, $val))])
    info
```

После сохранения изменений собираем пакет заново при помощи `Pkg.build("Match")`, и теперь он будет работать без назойливых предупреждений. После сборки следует обновить базу пакетов командой `Pkg.update()`. Кроме того, следует регулярно обновлять всю базу пакетов на предмет каких-либо изменений. В редких случаях, для того чтобы вновь установленный пакет определился в среде Julia, следует закрыть окно консоли и открыть его заново.

Перед началом использования пакета `PyPlot` в окне консоли интерпретатора REPL при помощи `Pkg.status()` следует проверить его наличие и установить, если его нет, используя `Pkg.add("PyPlot")`. Неплохо также обновить или установить браузер Chrome или Firefox. По отзывам, они хорошо ладят с `IJulia`. Более того, у них имеются расширения для работы с файлами записных книжек `.ipynb`. При первом использовании пакета оператор `using PyPlot` выполнит его инициализацию, что потребует некоторого времени, при последующих запусках визуализация выполняется почти мгновенно.

Под Windows 8 пакет `Winston` может сразу не заработать, но скорее всего эта проблема станет неактуальной в связи с обновлениями. И тем не менее проблема связана с конфликтом версий, в частности, версий 8.6 и 8.5, библиотек Tk/Tcl (скорее всего, вызвана установленным в системе дистрибутивом Anaconda Python). Чтобы решить проблему, нужно перейти в папку `.julia` (`C:/Users/ИМЯ_ПОЛЬЗОВАТЕЛЯ/.julia/`), перейти на 3 уровня ниже (`[VERSION]/Tk/deps/`) и открыть в текстовом редакторе файл `build.jl`. Перейти в раздел зависимостей (`dependencies`) в самом начале файла и отредактировать 2 строки, удалив все, что имеет цифры 8.6 или 86, например, `"tcl86t"`. Это касается зависимостей как для `tcl`, так и для `tk`. Затем перезапустить Julia и собрать Tk заново: `Pkg.build("Tk")`. Если проблемы продолжатся, то переустановите Julia повторно.

Вообще, желательно сперва установить дистрибутив Anaconda Python и только потом Julia.

Установка и удаление IDE Juno/Atom

В Windows следует скачать установщик универсального редактора программного кода Atom (<https://atom.io/>) и запустить его на выполнение.

Установка редактора Atom в Linux/Ubuntu через PPA выполняется следующим образом.

```
sudo add-apt-repository ppa:webupd8team/atom
```

```
sudo apt-get update
sudo apt-get install atom
```

Команды удаления IDE Atom/Juno:

```
sudo apt-get remove atom
sudo add-apt-repository --remove ppa:webupd8team/atom
```

Процедура настройки IDE Juno/Atom одинакова как для Windows, так и для Linux/Ubuntu (не забываем добавить соответствующий пакет в среду разработки Julia — `Pkg.add("Atom")`):

1. Запустить Atom и открыть окно настроек. Для этого выбрать элемент меню *File - Settings* и в окне настроек выбрать установку расширений (*+Install*).
2. Установить пакет-надстройку `uber-juno` набрав в поле ввода его название (Juno работает поверх редактора Atom) и нажав кнопку *Install*. Он настроит среду Juno, загрузив ряд других пакетов, при этом может понадобиться вручную удалить и заново установить пакет `ink`.
3. Факультативно, чтобы получить LaTeX-подобное автодополнение, установите пакет `latex-completions`. Тогда, например, запись `\alpha<TAB>` будет генерировать символ Юникода α .

Установка и работа с плагином для Eclipse

Скачать интерактивную среду разработки Eclipse для разработчиков на Java **Eclipse IDE for Java Developers** (<https://www.eclipse.org/downloads/>) в версии 4.5 или выше. Например, `eclipse-java-mars-2-win32-x86_64.zip`, и просто распаковать архивный файл, после чего он будет готов к работе.

Затем скачать плагин JuliaDT с сайта [JuliaComputing.com](http://juliacomputing.com) (<http://juliacomputing.com/blog/2016/02/06/Eclipse-JuliaDT.html>). На сегодняшний день имеются два релиза плагина: `v0.0.1` и `v0.0.2` (<https://github.com/JuliaComputing/JuliaDT/releases/tag/v0.0.1>, [.../v0.0.2](https://github.com/JuliaComputing/JuliaDT/releases/tag/v0.0.2))

Для установки плагина в Eclipse выбрать `Help > Install New Software...` Нажать `Add...` Нажать `Archive` и выбрать предварительно загруженный с сайта зашифрованный файл плагина. Набрать `JuliaDT` в поле для названия ПО и затем нажать `Next > Next`, подтвердить лицензионное соглашение `> Finish` и в конце перезагрузить Eclipse (инструкция по установке приведена на странице <https://github.com/JuliaComputing/JuliaDT/wiki/Installation>)

Затем следует интегрировать в среду Eclipse интерпретатор Julia. Для этого необходимо выполнить следующее (подразумевается, что Julia уже установлен):

В Eclipse выбрать `Preferences > Julia > Interpreters`. Затем нажать `Add...`, затем нажать `Browse...` Выбрать исполнимый файл Julia, расположенный в подпапке

/bin, затем нажать Ok и в конце еще раз Ok.

Для создания проектов на Julia следует выбрать File > New > Project > Julia > Empty Julia Project, нажать Next, затем в поле Project name ввести название проекта и в конце нажать Finish (инструкция по работе приведена на странице <https://github.com/JuliaComputing/JuliaDT/wiki/Use-Guide>).

Для выполнения программы на Julia надо в навигаторе пакетов Package Explorer выбрать нужный файл с программой, правой кнопкой вызвать контекстное меню и в нем выбрать Run as > REPL Julia Script. Результаты выполнения программы будут выведены во встроенную в Eclipse консоль либо для графиков и изображений отражены в отдельном окне или в браузере.

Работа с Julia в облаке JuliaBox

Чтобы начать работу в облаке JuliaBox необходимо сначала зарегистрироваться; можно воспользоваться существующей учетной записью Google+. Для загрузки файла (записной книжки) в облако, нужно нажать на значке файлового менеджера *Files*, и в открывшемся окне перетащить файл с расширением `.ipynb` или `.jl` из папки на вашем компьютере в зону для перетаскивания *Drag and Drop*.

Данная веб-служба также предоставляет веб-консоль для написания и выполнения несложных фрагментов кода. Для этого возле подсказки `juser@juliabox:~$` нужно набрать `julia`, после чего консоль перейдет в нормальный режим ожидания команд, как в обычной локальной консоли интерпретатора REPL. Правда, следует учесть, что в консоли используется слегка устаревшая версия языка.

Далее приведены процедуры установки и удаления ряда компонентов в операционных системах Windows и Linux/Ubuntu. Соответствующие процедуры для Mac OS X здесь не рассматриваются.

Установка и удаление дистрибутива Anaconda Python

Anaconda — это полностью свободный дистрибутив Python (предназначенный в том числе для коммерческого использования и повторного распространения). Он содержит более 400 самых популярных библиотек Python для вычислений в области естественных наук, математики, инженерии и анализа данных.

Установка дистрибутива в Windows выполняется стандартным образом после загрузки с сайта бинарного файла дистрибутива. В Linux сначала необходимо скачать установщик — скриптовый файл с расширением `.sh` для оболочки Bash (https://www.continuum.io/downloads#_unix). На примере Anaconda Python версии 3 (для Python 3.5) команда установки выглядит так: `bash Anaconda3-4.0.0-Linux-x86_64.sh`

Отметим, что инсталляция вступит в силу после того, как вы закроете и заново

откроете окно терминала.

Для обновления дистрибутива Anaconda Python, следует набрать в терминале следующую команду:

```
conda update conda
```

Удаление дистрибутива Anaconda:

```
rm -rf ~/anaconda
```

Более подробная информация по деинсталляции Anaconda содержится по указанной выше ссылке.

Чтобы удостовериться, что дистрибутив Anaconda установлен успешно, воспользуйтесь следующей командой:

```
conda --version
```

В результате будет выведен номер установленной версии.

Запуск сервера записных книжек Jupyter

В Windows и Linux/Ubuntu локальный сервер записных книжек запускается из командной строки двумя способами. Для этого открываем окно командной оболочки (cmd) в Windows или окно терминала в Linux/Ubuntu и просто набираем:

```
jupyter notebook
```

Или, как вариант:

```
ipython notebook --profile julia
```

Теперь интерактивная вычислительная среда IPython — это составная часть интегрированной среды Jupyter, и поэтому второй вариант запуска не рекомендован к применению и будет в будущем удален, к тому же в нем используется формат записной книжки предыдущей версии 3.0 (текущая 4.0). Запуск из консоли интерпретатора REPL в обоих случаях одинаков:

```
using IJulia; notebook()
```

Факультативно

Установка инструментальной среды Spyder

Spyder — это инструментальная среда для научных вычислений для языка Python (Scientific Python Development Environment) для Windows, Mac OS X и Linux. Это простая, легковесная и бесплатная интерактивная среда разработки

на Python, которая предлагает функционал, аналогичный среде разработки на MATLAB, включая готовые к использованию виджеты PyQt4 и PySide: редактор исходного кода, редактор массивов данных NumPy, редактор словарей, консоли Python и IPython и многое другое.

Для пользователей Windows хорошая новость заключается в том, что Spyder уже включен в состав дистрибутива Anaconda Python, и исполнимый файл находится в папке `C:\Users\[ИМЯ_ПОЛЬЗОВАТЕЛЯ]\Anaconda3\Scripts\spyder.exe`. Чтобы установить среду Spyder в Linux/Ubuntu, используя официальный менеджер пакетов, нужна всего одна команда:

```
sudo apt-get install spyder
```

Чтобы установить с использованием менеджера пакетов pip:

```
sudo apt-get install python-qt4 python-sphinx  
sudo pip install spyder
```

И чтобы обновить:

```
sudo pip install -U spyder
```

Управление пакетами .deb в Linux/Ubuntu

Установка пакетов .deb в терминале (пакет должен находиться под корневым каталогом home)

```
sudo dpkg -i [пакет].deb
```

или

```
sudo dpkg --install [пакет].deb
```

Удаление пакетов .deb

```
sudo dpkg -r [пакет].deb
```

Удаление вместе с конфигурационными файлами:

```
sudo dpkg -P [пакет].deb
```