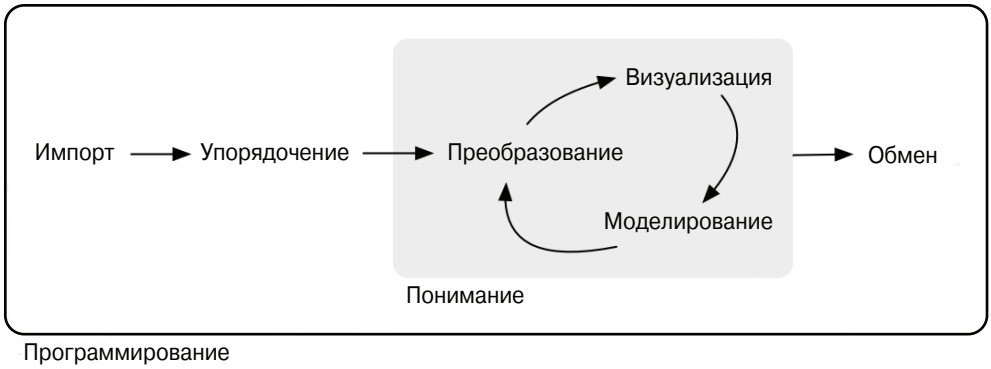
Предисловие

Наука о данных — увлекательная дисциплина, позволяющая превращать необработанные первичные данные в гипотезы и новые знания. Цель книги — помочь вам в изучении наиболее важных средств R, освоив которые вы сможете приступить к профессиональной деятельности в этой области. Прочитав книгу, вы научитесь работать с инструментами, обеспечивающими эффективное применение языка R для решения широкого круга задач науки о данных.

О чем эта книга

Наука о данных — обширная область, для освоения которой недостаточно прочтения только одной книги. Задача этой книги — помочь вам как можно быстрее овладеть наиболее важными инструментами. Наше видение того, какие инструментальные средства могут понадобиться вам в типичном проекте науки о данных, можно проиллюстрировать следующей схемой.



Прежде всего вам потребуется *импортировать* данные в вычислительную среду R. В типичных случаях данные берутся из файла, базы данных или сайта и загружаются во фрейм данных R. Не имея возможности получить свои данные в R, вы не сможете их исследовать!

Как правило, импортированные данные целесообразно предварительно *упорядочить*, т.е. привести к определенному аккуратному виду. Аккуратно организованные данные — это данные, сохраненные в таком виде, который обеспечивает согласованность семантики набора данных со способом их хранения. Другими словами, если ваши данные аккуратно организованы, или, как говорят, *опрятны* (tidy), то каждый столбец представляет переменную, а каждая строка — наблюдение. Понимаемая в этом смысле опрятность данных имеет большое значение, поскольку следование единому принципу их структуризации позволяет вам не тратить каждый раз усилия на приведение их к виду, необходимому для использования совместно с той или иной функцией, а сосредоточиться на задачах, касающихся данных как таковых.

Обычно, после того как данные приведены к аккуратному виду, первое, что необходимо сделать, — это *преобразовать* их. Преобразование данных включает сужение их до набора интересующих вас наблюдений (например, путем отбора только данных, относящихся к населению одного города, или данных за последний год), создание новых переменных, являющихся функциями существующих переменных (например, вычисление скорости по известным расстоянию и времени), и вычисление вспомогательных статистик (например, количества элементов или их среднего). Объединенный процесс приведения данных к аккуратному виду и их преобразования называют *подготовкой* данных.

Для генерации знаний на основе надлежащим образом подготовленных данных существуют два основных механизма: визуализация и моделирование. У каждого из них есть свои сильные и слабые стороны, но они взаимно дополняют друг друга, поэтому при выполнении практического анализа вы будете не один раз попеременно применять каждый из них.

Визуализация информации — важный вид человеческой деятельности. Хорошая визуализация обнажит те стороны явления, которые вы даже не ожидали увидеть, или заставит поставить новые вопросы в отношении данных. Кроме того, хорошая визуализация может подсказать вам, что вы ставите не те вопросы или что требуется сбор других данных. Визуализация способна удивить вас своими результатами, но соответствующие решения масштабируются не слишком хорошо, поскольку интерпретация результатов требует вмешательства человека.

Модели дополняют визуализацию. Если вы достаточно точно сформулируете свои вопросы, то модель поможет вам найти ответы на них. В своей основе модели служат математическим или вычислительным инструментом,

поэтому, как правило, проблем с масштабированием не возникает. Даже если проблемы и появляются, то обычно они решаются покупкой дополнительных вычислительных мощностей. (Мозги ведь все равно не купишь!) Однако любая модель основана на определенных допущениях, которые она сама не может подвергать сомнению. Это означает, что в основном модель ничем не сможет вас удивить.

Завершающий этап науки о данных — это *обмен информацией*, являющийся неотъемлемой частью любого аналитического проекта. Какого бы глубокого понимания данных вы ни достигли с помощью своих моделей и визуализации, это не будет иметь ровным счетом никакого значения, если вы не сможете поделиться результатами с другими людьми.

Программирование играет роль объединяющего начала по отношению ко всем описанным инструментам. Программирование — это всепроникающий инструмент, который можно использовать в любой части проекта. Чтобы заниматься наукой о данных, вовсе не обязательно быть специалистом по программированию, но знание программирования никогда не помешает, поскольку это позволит вам автоматизировать рутинные виды работ и решать новые задачи с гораздо меньшими усилиями.

Описанные инструментальные средства будут использоваться вами в любом проекте, связанном с анализом данных, но чаще всего вы не сможете обойтись только ими. Здесь работает так называемое правило “80/20”: для выполнения примерно 80% объема работ любого проекта вы будете использовать инструменты, которые изучите в этой книге, но для выполнения оставшихся 20% работ вам потребуются другие инструменты. В необходимых случаях в книге даются ссылки на источники, в которых вы сможете узнать больше о дополнительных инструментах.

Как организована книга

Используемые в науке о данных инструменты были описаны выше примерно в том порядке, в каком они будут применяться по мере выполнения анализа (хотя, разумеется, вы будете неоднократно переходить от одного из них к другому). Однако, как показывает наш опыт, изучение инструментов в такой очередности не является оптимальным, что обусловлено следующими причинами.

- Начинать с изучения инструментов, предназначенных для получения данных и приведения их к нужному виду, не совсем целесообразно,

поскольку 80% этой работы — сплошная рутина и скука, тогда как при выполнении остальных 20% вас не будет покидать чувство неопределенности и раздражения. Совершенно неподходящее начало для изучения нового предмета! Вместо этого мы начнем с визуализации и преобразования данных, которые уже импортированы и приведены к надлежащему виду. Благодаря этому самостоятельное выполнение операций импорта и приведения данных к аккуратной форме в качестве подготовки к анализу в будущем не сможет убить ваш энтузиазм, поскольку вы будете знать, что игра стоит свеч.

- Некоторые темы лучше всего объяснять с использованием других инструментальных средств. Например, мы считаем, что понять, как работают модели, вам будет легче, если перед этим вы познакомитесь с визуализацией, приведением данных к аккуратному виду и программированием.
- Средства программирования могут не представлять для вас самостоятельного интереса, но с их помощью вы сможете решать гораздо более широкий круг трудных задач. В середине книги мы познакомим вас с некоторыми программными инструментами, а затем покажем, как их сочетание с традиционными инструментами науки о данных позволяет справиться со многими интересными задачами моделирования.

В каждой главе мы старались следовать одному и тому же подходу: начинать с показательных примеров, чтобы вы могли увидеть общую картину, а затем углубляться в рассмотрение деталей. Каждый раздел книги сопровождается контрольными упражнениями, которые помогут вам закрепить на практике пройденный материал. Не поддавайтесь соблазну пропускать упражнения, поскольку нет лучшего способа обучения, чем решение реальных задач.

О чем не будет говориться в книге

Существует ряд важных тем, которые не рассматриваются в этой книге. Мы считаем исключительно важным сфокусировать ваше внимание исключительно на тех вещах, знание которых позволит вам приступить к самостоятельной работе в кратчайшие сроки. Это означает, что из поля нашего зрения выпадут некоторые важные темы, перечисленные ниже.

Большие данные

В этой книге мы ограничиваемся рассмотрением небольших, умещающихся в памяти компьютера наборов данных. Это неплохо для начала, поскольку справиться с данными большого объема, или, в соответствии с современной терминологией, с "большими данными" (Big Data), без предварительно приобретенного опыта работы с данными небольшого объема вы просто не сможете. Инструменты, которые вы изучите в книге, позволяют легко обрабатывать сотни мегабайт данных, а приложив дополнительные усилия, вы сможете увеличить этот объем до 1-2 Гбайт. Если обычно вам приходится работать с данными большего объема (скажем, 10-100 Гбайт), воспользуйтесь пакетом **data.table** (`https://github.com/Rdatatable/data.table`). В данной книге он не рассматривается ввиду предельной лаконичности его интерфейса, который предлагает слишком мало лингвистических подсказок, из-за чего его изучение представляет определенные трудности. Но если вы часто работаете с большими данными, то выигрыш в производительности стоит того, чтобы затратить усилия на изучение этого пакета.

В случае еще более крупных данных тщательно проанализируйте, не может ли быть так, что проблема больших данных на самом деле является замаскированной проблемой небольших данных. В то время как полный набор данных может быть большим, зачастую для получения ответа на конкретный вопрос достаточно использовать лишь небольшую порцию данных. Возможно, вам удастся найти поднабор меньших размеров, который умещается в памяти и при этом все еще позволяет получить ответ на интересующий вас вопрос. В таком случае наибольшую трудность представляет нахождение подходящего небольшого набора, на что часто требуется не одна попытка.

Другая возможность заключается в том, чтобы попытаться представить задачу с большими данными в виде множества задач меньшего масштаба. Каждая отдельная задача может легко помещаться в памяти, хотя их общее число может исчисляться миллионами. Например, первоначально вы строите модель для каждого объекта в своем наборе данных. Если объектов всего 10 или 100, то это тривиальная задача, но все значительно усложняется при наличии нескольких миллионов объектов. К счастью, каждая задача решается независимо от других (иногда такие ситуации описывают несколько двусмысленным термином *параллелизм*), поэтому вам достаточно воспользоваться одной из подходящих систем распределенных вычислений (например, Hadoop или Spark), позволяющих обрабатывать разные наборы данных

на разных компьютерах. Выяснив с помощью описанных в этой книге инструментов, каким может быть ответ на вопрос, относящийся к одиночному набору данных, вы можете изучить новые инструменты, такие как **sparklyr**, **rhipe** и **dd**, и использовать их для решения задачи, относящейся к полному набору данных.

Python, Julia и прочее

В книге ничего не говорится о Python, Julia или любом другом языке программирования из тех, которые применяются в науке о данных. И не потому, что каждый из этих языков плох в каком-либо отношении. Вовсе нет! На практике в большинстве случаев исследования проводятся с использованием одновременно нескольких языков, включая, как правило, R и Python.

Вместе с тем мы твердо убеждены, что лучше всего осваивать инструменты поочередно. Гораздо эффективнее глубоко изучить что-то одно, чем распылять силы на изучение сразу нескольких тем. Отсюда не следует, что вы должны знать только какой-то один предмет. Это лишь означает, что ваше обучение пройдет значительно быстрее, если вы будете каждый раз концентрироваться только на одном предмете. Вы должны стремиться узнавать новое на протяжении всей своей карьеры, но, прежде чем приступать к изучению очередной интересной темы, вы должны быть уверены в том, что твердо усвоили предыдущий материал.

Мы считаем изучение языка R отличной отправной точкой для увлекательного путешествия в мир науки о данных, поскольку он изначально проектировался для применения в данной области. R — не только язык программирования, но и интерактивная вычислительная среда для исследования данных. В отношении поддержки интерактивности R обладает значительно большей гибкостью по сравнению со многими из его аналогов. Эта гибкость сопряжена и с определенными недостатками, но они компенсируются легкостью введения грамматик, специально приспособленных для выполнения отдельных этапов исследовательского процесса. Эти мини-языки помогают вам мыслить категориями специалиста в области науки о данных и вместе с тем чрезвычайно облегчают ваше взаимодействие с компьютером.

Непрямоугольные данные

В книге рассматриваются исключительно прямоугольные данные: коллекции значений, каждое из которых связано с одной переменной и одним

наблюдением. Существует много наборов данных, не вписывающихся естественным образом в эту парадигму, к числу которых относятся изображения, звуковые данные, двоичные деревья и текст. Но прямоугольные фреймы данных широко распространены в науке и промышленности, и мы считаем, что ваше путешествие в мир науки о данных целесообразно начать именно с них.

Подтверждение гипотез

Процедуры анализа данных можно разделить на два класса: выдвижение гипотез и их подтверждение (также употребляются термины *разведочный анализ* и *подтверждающий анализ*). В этой книге рассматривается исключительно процедура выдвижения гипотез, т.е. разведочный анализ. В ходе анализа вы тщательно изучаете данные и, используя свои знания предметной области, выдвигаете ряд гипотез, предлагающих возможные объяснения того, почему данные ведут себя так, а не иначе. При этом вы оцениваете гипотезы неформально, с позиций здравого скептицизма, основанного на рассмотрении данных под разными углами.

Выдвижение гипотез должно дополняться попытками их подтверждения. Подтверждение гипотез затруднено следующими двумя обстоятельствами.

- Вы должны иметь в своем распоряжении точную математическую модель, позволяющую генерировать опровергаемые гипотезы. Зачастую это требует углубления в дебри статистики.
- Любое наблюдение можно использовать в целях подтверждения гипотезы только один раз. Использование наблюдения более одного раза немедленно возвращает вас обратно к разведочному анализу. Это означает, что еще до того, как вы приступите к процедуре подтверждения гипотезы, вы должны "зарегистрировать" (заранее подготовить в явном виде) свой план анализа и не отклоняться от него, когда ознакомитесь с данными. О выборе стратегий, которые можно использовать для упрощения этой задачи, речь пойдет в части IV.

Общепринято считать моделирование инструментом выдвижения гипотез, а визуализацию — инструментом их подтверждения. Но эта дихотомия не совсем верна: модели часто применяют на этапе подтверждающего анализа, а визуализацию — в разведочном анализе. Ключевым фактором их дифференциации является частота использования каждого наблюдения: если наблюдение рассматривается только один раз, то вы имеете дело с подтверждением

гипотезы; если же наблюдение рассматривается более одного раза, то это означает, что выполняется разведочный анализ.

Что необходимо для работы с книгой

Чтобы чтение книги принесло вам максимальную пользу, установите на своем компьютере описанные ниже ресурсы. Мы рассчитываем на то, что вы математически грамотны; наличие предварительного опыта программирования будет только плюсом. Если до этого вы никогда не занимались программированием, то, возможно, вас заинтересует книга Гарретта *Hands-On Programming with R*, которая отлично дополнит данную книгу.

Для выполнения приведенных в книге примеров вам понадобятся следующие средства: R, RStudio, коллекция пакетов R в виде библиотеки *tidyverse* и довольно большое количество других пакетов, включающих готовые функции, документацию с их описанием и образцы данных.

R

Загрузите R из сети CRAN (от англ. *comprehensive R archive network*). Эта сеть состоит из множества серверов-зеркал, разбросанных по всему миру, и используется для распространения R и пакетов R. Не пытайтесь определить, какой из серверов является для вас ближайшим: просто воспользуйтесь зеркальным облаком `https://cloud.r-project.org`, которое сделает это автоматически.

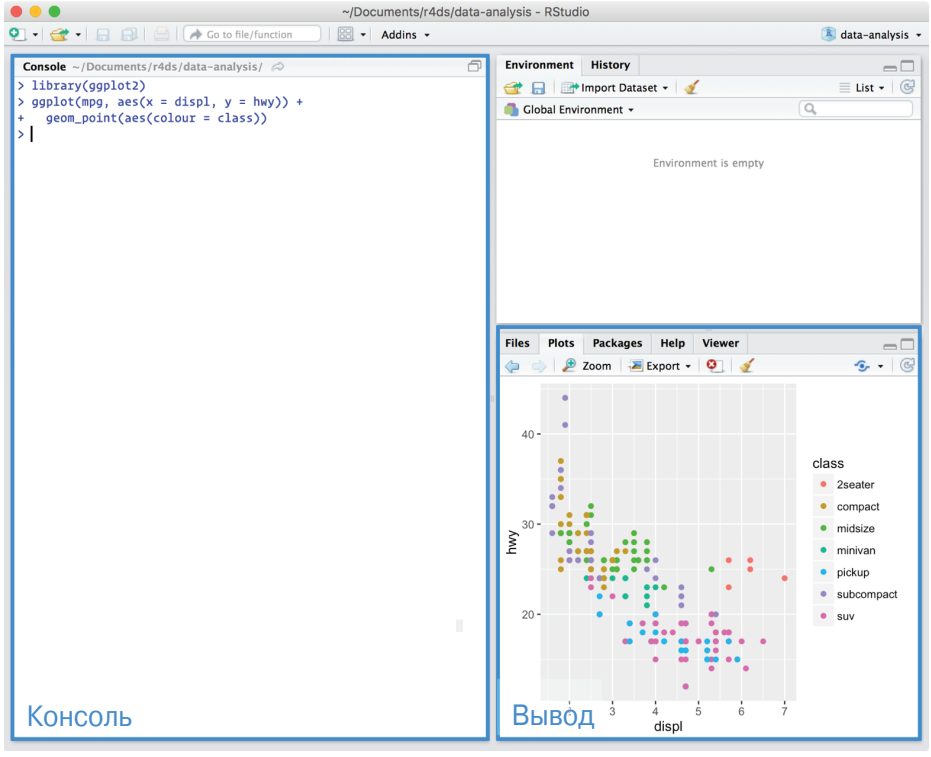
Каждый год выпускаются одна главная версия R и 2-3 вспомогательные. Целесообразно регулярно обновлять установленную у вас версию. Обновление, особенно в случае главной версии, может доставить определенные неудобства, поскольку потребует переустановки всех пакетов, но отказываться не стоит, ведь лучше идти в ногу с прогрессом.

RStudio

RStudio — это интегрированная среда разработки (IDE) для программирования на R. Чтобы загрузить и установить среду, посетите сайт `http://www.rstudio.com/download`. RStudio обновляется пару раз в год. Если доступна очередная версия, RStudio известит вас об этом. Имеет смысл регулярно обновлять программу, поскольку в таком случае вы сможете воспользоваться

преимуществами новых средств. Для работы с книгой вам потребуется версия RStudio не ниже 1.0.0.

Запустив RStudio, вы увидите окно, разделенное на две основные области.



Пока что вам достаточно знать, что код следует вводить в панели консоли и запускать нажатием клавиши <Enter>. Остальные подробности будут сообщены позже.

Библиотека `tidyverse`

Вам также потребуется установить некоторые из пакетов R. Пакет R — это коллекция функций, данных и документации, расширяющая возможности базовой комплектации R. Использование пакетов является залогом успешной работы с R. Бо́льшая часть пакетов, которые вы изучите в книге, входит в состав библиотеки *tidyverse*. Пакеты *tidyverse* разделяют общую философию обработки данных и программирования на языке R и были спроектированы так, чтобы обеспечить их естественное взаимодействие.

Для установки полной библиотеки *tidyverse* понадобится всего одна строка кода:

```
install.packages("tidyverse")
```

Введите этот код в окне консоли и выполните его, нажав клавишу <Enter>. R загрузит пакеты из CRAN и установит их на вашем компьютере. Если в процессе установки возникнут проблемы, убедитесь в том, что вы подключены к Интернету, а адрес `https://cloud.r-project.org/` не заблокирован вашим брандмауэром или прокси-сервером.

Вы не сможете использовать функции, объекты и файлы справки пакета, если не загрузите его с помощью функции `library()`. Это можно сделать в любой момент после установки пакета.

```
library(tidyverse)
#> Loading tidyverse: ggplot2
#> Loading tidyverse: tibble
#> Loading tidyverse: tidyr
#> Loading tidyverse: readr
#> Loading tidyverse: purrr
#> Loading tidyverse: dplyr
#> Conflicts with tidy packages ----------------------------------
#> filter(): dplyr, stats
#> lag():    dplyr, stats
```

Сообщения уведомляют вас о загрузке пакетов **ggplot2**, **tibble**, **tidyr**, **readr**, **purrr** и **dplyr**. Эти пакеты считаются ядром библиотеки *tidyverse*, поскольку вы будете использовать их при выполнении практически любого вида анализа.

Пакеты библиотеки *tidyverse* обновляются довольно часто. Чтобы узнать о наличии обновлений и, по желанию, установить их, воспользуйтесь командой `tidyverse_update()`.

Другие пакеты

Существует множество великолепных пакетов, не являющихся частью библиотеки *tidyverse*, поскольку они предназначены для решения задач, относящихся к другим предметным областям, или проектировались на основе других базовых принципов. Это не делает их лучше или хуже — они просто другие. По мере того как количество ваших проектов, выполняемых в R, будет расти, вы будете знакомиться с новыми пакетами и новыми способами исследования данных.

В этой книге используются три пакета, не входящих в состав библиотеки *tidyverse*:

```
install.packages(c("nycflights13", "gapminder", "Lahman"))
```

Указанные пакеты предоставляют данные об авиарейсах, уровне развития различных стран и бейсбольной статистике. Мы будем использовать для иллюстрации основных идей науки о данных.

Выполнение R-кода

В предыдущем разделе уже было приведены некоторые примеры выполнения R-кода. В книге код выглядит примерно так.

```
1 + 2
#> [1] 3
```

Если вы выполните этот же код в окне консоли, то он будет выглядеть несколько иначе:

```
> 1 + 2 [1] 3
```

Здесь можно выделить два основных отличия. В окне консоли ваш ввод начинается после символа `>`, так называемого *приглашения* (мы не показываем его в книге). В книжном варианте вывод превращен в комментарии с помощью символов `#>`; в окне консоли вывод появляется непосредственно вслед за кодом. Эти два отличия означают, что если вы работаете с электронной версией книги, то можете свободно копировать код из книги и вставлять его на консоль.

На протяжении всей книги мы придерживаемся следующих соглашений при ссылках на код:

- для имен функций используется моноширинный шрифт, и за каждым таким именем следует пара скобок, например `sum()` или `mean()`;
- для других объектов R (таких, как данные или аргументы функций) также используется моноширинный шрифт, но их имена указываются без круглых скобок, например `flights` или `x`;
- если мы хотим уточнить, к какому пакету относится объект, то перед именем объекта указывается имя пакета, за которым следует двоеточие (например, `dplyr::mutate()` или `nycflights13::flights`); это тоже действительный R-код.

Получение справочной и дополнительной информации

Эта книга не является исчерпывающим пособием, да и не существует такой книги, прочитав которую вы смогли бы сказать, что полностью освоили R. По мере того как вы начнете применять описанные в данной книге методы в своих проектах, вы очень быстро столкнетесь с вопросами, на которые мы не даем ответа. Ниже вы найдете некоторые советы относительно того, где можно получить необходимые справочные сведения и как организовать свое дальнейшее обучение.

Зайдя в тупик, попытайтесь выбраться из него с помощью Google. Обычно добавления "R" в конце запроса достаточно для того, чтобы ограничить круг выводимых результатов поиска до нужного вам; если такой поиск окажется бесполезным, то это будет означать, что специфические для R результаты поиска отсутствуют. Если вы получаете сообщение об ошибке и не можете понять, что оно означает, попытайтесь "погуглить" его! Существует вероятность того, что кто-то другой ранее уже интересовался этим вопросом и соответствующая информация доступна где-то в Интернете. (Если сообщение об ошибке выведено не на английском языке, выполните команду `Sys.setenv(LANGUAGE = "en")` и перезапустите код; скорее всего, вам удастся получить справку об этом сообщении на английском языке.)

Если вам не поможет Google, попытайтесь найти помощь на сайте Stackoverflow (`http://stackoverflow.com/`). Для начала потратьте некоторое время на поиск ответа, который кто-то уже мог предложить; включение *[R]* в запрос позволяет сузить круг искомых вопросов и ответов до тех, которые имеют отношение к R. Если ничего полезного найти не удается, подготовьте воспроизводимый пример минимального объема. Увидев этот пример, другим людям будет легче оказать вам помощь, а нередко в процессе подготовки такого примера решение проблемы удается найти самому.

Чтобы сделать пример воспроизводимым, включите в него требуемые пакеты, данные и код.

- *Пакеты* должны загружаться в самом начале скрипта, чтобы можно было сразу же увидеть, какие из них используются в примере. Это наиболее подходящий момент убедиться в том, что вы используете самые последние версии каждого из пакетов. Вполне возможно, что вы столкнулись с ошибкой, которая была устранена с того момента, когда вы в последний раз устанавливали этот пакет на своем компьютере. В случае

пакетов библиотеки *tidyverse* самым простым способом проверки является выполнение функции `tidyverse_update()`.

- Самый простой способ включения в пример используемых данных — это вызов функции `dput()` для генерации кода, воспроизводящего данные. Например, чтобы воссоздать в R набор данных `mtcars`, поступите следующим образом:

1) вызовите в R функцию `dput(mtcars)`;
2) скопируйте вывод;
3) введите в своем воспроизводимом скрипте строку `mtcars <-`, а затем вставьте содержимое буфера обмена.

Попытайтесь найти минимальный набор данных, при котором проблема все еще проявляется.

- Не пожалейте времени на то, чтобы привести код к виду, в котором его будет легко читать другим людям:
  - обязательно используйте пробелы, облегчающие чтение текста, и короткие, но информативные имена переменных;
  - используйте комментарии для указания проблемных мест в коде;
  - исключите из примера все то, что не имеет отношения к возникшей проблеме.

  Чем короче код, тем он понятнее и тем легче его исправить.

Завершите этот процесс проверкой того, что вы действительно подготовили воспроизводимый пример, начав в R новый сеанс и скопировав в него свой скрипт.

Кроме того, постоянно готовьтесь к самостоятельному устранению проблем, прежде чем они возникнут. Ежедневное изучение R небольшими порциями окупится сторицей в долгосрочной перспективе. Следите за сообщениями Хэдли, Гарретта и других сотрудников компании RStudio, публикуемыми в блоге (`https://blog.rstudio.org/`). Именно там мы размещаем объявления о новых пакетах, новых возможностях IDE и персональных курсах. Возможно, вы захотите следить в Твиттере за сообщениями Хэдли (`@hadleywickham`) или Гарретта (`@statgarrett`) или быть в курсе всех новинок IDE (`@rstudiotips`).

Для поддержания контакта с более обширным R-сообществом рекомендуем посещать сайт `http://www.r-bloggers.com`: он агрегирует более 500

блогов со всего света по тематике R. Если вы активный пользователь Твиттера, следите за хеш-тегом #rstats. Твиттер — один из основных инструментов, используемых Хэдли для того, чтобы постоянно быть в курсе новых разработок, появляющихся в сообществе.

Электронная версия книги

Электронная версия книги на английском языке доступна на сайте http://r4ds.had.co.nz. Она будет непрерывно обновляться в промежутках между выпусками дополнительных тиражей печатного издания. Исходный код книги доступен по адресу https://github.com/hadley/r4ds. Книга поддерживается сайтом https://bookdown.org, который упрощает преобразование файлов в формате R Markdown в файлы формата HTML, PDF и EPUB.

Эта книга создавалась с использованием следующих ресурсов.

```
devtools::session_info(c("tidyverse"))
#> Session info -------------------------------------------------
#>  setting  value
#>  version  R version 3.3.1 (2016-06-21)
#>  system   x86_64, darwin13.4.0
#>  ui       X11
#>  language (EN)
#>  collate  en_US.UTF-8
#>  tz       America/Los_Angeles
#>  date     2016-10-10
#> Packages -----------------------------------------------------
#>  package      * version    date       source
#>  assertthat     0.1        2013-12-06 CRAN (R 3.3.0)
#>  BH             1.60.0-2   2016-05-07 CRAN (R 3.3.0)
#>  broom          0.4.1      2016-06-24 CRAN (R 3.3.0)
#>  colorspace     1.2-6      2015-03-11 CRAN (R 3.3.0)
#>  curl           2.1        2016-09-22 CRAN (R 3.3.0)
#>  DBI            0.5-1      2016-09-10 CRAN (R 3.3.0)
#>  dichromat      2.0-0      2013-01-24 CRAN (R 3.3.0)
#>  digest         0.6.10     2016-08-02 CRAN (R 3.3.0)
#>  dplyr        * 0.5.0      2016-06-24 CRAN (R 3.3.0)
#>  forcats        0.1.1      2016-09-16 CRAN (R 3.3.0)
#>  foreign        0.8-67     2016-09-13 CRAN (R 3.3.0)
#>  ggplot2      * 2.1.0.9001 2016-10-06 local
#>  gtable         0.2.0      2016-02-26 CRAN (R 3.3.0)
#>  haven          1.0.0      2016-09-30 local
#>  hms            0.2-1      2016-07-28 CRAN (R 3.3.1)
#>  httr           1.2.1      2016-07-03 cran (@1.2.1)
#>  jsonlite       1.1        2016-09-14 CRAN (R 3.3.0)
```

```
#>  labeling       0.3        2014-08-23 CRAN (R 3.3.0)
#>  lattice        0.20-34    2016-09-06 CRAN (R 3.3.0)
#>  lazyeval       0.2.0      2016-06-12 CRAN (R 3.3.0)
#>  lubridate      1.6.0      2016-09-13 CRAN (R 3.3.0)
#>  magrittr       1.5        2014-11-22 CRAN (R 3.3.0)
#>  MASS           7.3-45     2016-04-21 CRAN (R 3.3.1)
#>  mime           0.5        2016-07-07 cran (@0.5)
#>  mnormt         1.5-4      2016-03-09 CRAN (R 3.3.0)
#>  modelr         0.1.0      2016-08-31 CRAN (R 3.3.0)
#>  munsell        0.4.3      2016-02-13 CRAN (R 3.3.0)
#>  nlme           3.1-128    2016-05-10 CRAN (R 3.3.1)
#>  openssl        0.9.4      2016-05-25 cran (@0.9.4)
#>  plyr           1.8.4      2016-06-08 cran (@1.8.4)
#>  psych          1.6.9      2016-09-17 CRAN (R 3.3.0)
#>  purrr        * 0.2.2      2016-06-18 CRAN (R 3.3.0)
#>  R6             2.1.3      2016-08-19 CRAN (R 3.3.0)
#>  RColorBrewer   1.1-2      2014-12-07 CRAN (R 3.3.0)
#>  Rcpp           0.12.7     2016-09-05 CRAN (R 3.3.0)
#>  readr        * 1.0.0      2016-08-03 CRAN (R 3.3.0)
#>  readxl         0.1.1      2016-03-28 CRAN (R 3.3.0)
#>  reshape2       1.4.1      2014-12-06 CRAN (R 3.3.0)
#>  rvest          0.3.2      2016-06-17 CRAN (R 3.3.0)
#>  scales         0.4.0.9003 2016-10-06 local
#>  selectr        0.3-0      2016-08-30 CRAN (R 3.3.0)
#>  stringi        1.1.2      2016-10-01 CRAN (R 3.3.1)
#>  stringr        1.1.0      2016-08-19 cran (@1.1.0)
#>  tibble       * 1.2        2016-08-26 CRAN (R 3.3.0)
#>  tidyr        * 0.6.0      2016-08-12 CRAN (R 3.3.0)
#>  tidyverse    * 1.0.0      2016-09-09 CRAN (R 3.3.0)
#>  xml2           1.0.0.9001 2016-09-30 local
```

Соглашения, принятые в книге

В книге приняты следующие типографские соглашения.

- *Курсивом* выделяются новые термины, имена и расширения имен файлов, названия библиотек.
- **Полужирным** шрифтом выделяются названия пакетов R.
- `Моноширинный` шрифт используется в листингах программ, а также в основном тексте для представления таких программных элементов, как названия функций и переменных, баз данных, типов данных, инструкций, ключевых слов, а также URL-адресов, адресов электронной почты и других подобных элементов.

- **`Полужирный моноширинный`** шрифт используется для представления команд и другого текста, который должен вводиться пользователем.
- *`Курсивный моноширинный`* шрифт используется для представления текста, который должен быть заменен вводимыми пользователем значениями, определяемыми контекстом.

Этой пиктограммой обозначены абзацы, содержащие советы и рекомендации.

Использование примеров кода

Исходный код всех примеров доступен по адресу `https://github.com/hadley/r4ds`.

Книга была написана для того, чтобы облегчить вам работу. Вообще говоря, вы можете свободно использовать приведенный в книге код в своих программах и документации. Получения какого-либо специального разрешения от нас, если только речь не идет о значительных объемах кода, не требуется. Например, использование в вашей программе нескольких фрагментов кода, взятых из этой книги, не требует разрешения. Однако продажа или распространение компакт-диска, содержащего примеры из книг, выпущенных издательством O'Reilly, без предварительного получения разрешения запрещена. Цитирование данной книги и использование примеров кода в ответах на вопросы не требует разрешения. Вместе с тем, если вы включаете в документацию своего продукта значительные объемы кода из приведенных в книге примеров, то получение соответствующего разрешения является обязательным условием.

Ждем ваших отзывов!

Вы, читатель этой книги, и есть главный ее критик. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что еще вы хотели бы увидеть изданным нами. Нам интересны любые ваши замечания в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать нам бумажное или электронное письмо либо просто посетить наш сайт и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится ли вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

Отправляя письмо или сообщение, не забудьте указать название книги и ее авторов, а также свой обратный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию новых книг.

Наши электронные адреса:

E-mail: info@dialektika.com

WWW: http://www.dialektika.com

Наши почтовые адреса:

в России: 195027, Санкт-Петербург, Магнитогорская ул., д. 30, ящик 116

в Украине: 03150, Киев, а/я 152