
Оглавление

Об авторе	13
Об иллюстрации на обложке	14
Предисловие от издательства	15
Предисловие	16
Графические выделения	16
О примерах кода	17
Как с нами связаться	17
Благодарности	18
Глава 1. Предварительные сведения.....	22
1.1. О чем эта книга?.....	22
Какого рода данные?	22
1.2. Почему именно Python?	23
Python как клей.....	23
Решение проблемы «двух языков»	24
Недостатки Python.....	24
1.3. Необходимые библиотеки для Python	25
NumPy.....	25
pandas	25
matplotlib.....	27
IPython и Jupyter	27
SciPy.....	28
scikit-learn	28
statsmodels	29
1.4. Установка и настройка.....	29
Miniconda в Windows	30
GNU/Linux.....	30
Miniconda в macOS.....	31
Установка необходимых пакетов	32
Интегрированные среды разработки (IDE)	33
1.5. Сообщество и конференции.....	33
1.6. Структура книги.....	34
Примеры кода	35
Данные для примеров	35
Соглашения об импорте.....	36

Глава 2. Основы языка Python, IPython и Jupyter-блокноты..... 37

2.1. Интерпретатор Python.....	38
2.2. Основы IPython	39
Запуск оболочки IPython.....	39
Запуск Jupyter-блокнота.....	40
Завершение по нажатию клавиши Tab.....	43
Интроспекция.....	45
2.3. Основы языка Python.....	46
Семантика языка	46
Скалярные типы	53
Поток управления.....	61
2.4. Заключение	64

Глава 3. Встроенные структуры данных, функции и файлы..... 65

3.1. Структуры данных и последовательности	65
Кортеж	65
Список	68
Словарь.....	72
Множество.....	76
Встроенные функции последовательностей	78
Списковое, словарное и множественное включения.....	80
3.2. Функции.....	82
Пространства имен, области видимости и локальные функции	83
Возврат нескольких значений	84
Функции являются объектами.....	85
Анонимные (лямбда-) функции	87
Генераторы.....	87
Обработка исключений.....	90
3.3. Файлы и операционная система	92
Байты и Unicode в применении к файлам	96
3.4. Заключение	98

Глава 4. Основы NumPy: массивы и векторные вычисления 99

4.1. NumPy ndarray: объект многомерного массива.....	101
Создание ndarray	102
Тип данных для ndarray	104
Арифметические операции с массивами NumPy.....	107
Индексирование и вырезание	108
Булево индексирование	113
Прихотливое индексирование.....	116
Транспонирование массивов и перестановка осей	117
4.2. Генерирование псевдослучайных чисел.....	119
4.3. Универсальные функции: быстрые поэлементные операции над массивами	120
4.4. Программирование на основе массивов.....	123
Запись логических условий в виде операций с массивами.....	125

Математические и статистические операции.....	126
Методы булевых массивов.....	128
Сортировка.....	128
Устранение дубликатов и другие теоретико-множественные операции	130
4.5. Файловый ввод-вывод массивов	130
4.6. Линейная алгебра.....	131
4.7. Пример: случайное блуждание.....	133
Моделирование сразу нескольких случайных блужданий	135
4.8. Заключение	136
Глава 5. Первое знакомство с pandas	137
5.1. Введение в структуры данных pandas	138
Объект Series	138
Объект DataFrame	142
Индексные объекты.....	149
5.2. Базовая функциональность.....	151
Переиндексация	151
Удаление элементов из оси.....	154
Доступ по индексу, выборка и фильтрация	155
Арифметические операции и выравнивание данных.....	165
Применение функций и отображение	170
Сортировка и ранжирование	172
Индексы по осям с повторяющимися значениями	175
5.3. Редукция и вычисление описательных статистик.....	177
Корреляция и ковариация	180
Уникальные значения, счетчики значений и членство.....	181
5.4. Заключение	185
Глава 6. Чтение и запись данных, форматы файлов.....	186
6.1. Чтение и запись данных в текстовом формате.....	186
Чтение текстовых файлов порциями	193
Вывод данных в текстовом формате.....	195
Обработка данных в других форматах с разделителями.....	196
Данные в формате JSON.....	198
XML и HTML: разбор веб-страниц.....	200
6.2. Двоичные форматы данных.....	203
Формат HDF5.....	205
6.3. Взаимодействие с HTML и Web API	208
6.4. Взаимодействие с базами данных	209
6.5. Заключение	211
Глава 7. Очистка и подготовка данных.....	212
7.1. Обработка отсутствующих данных	212
Фильтрация отсутствующих данных	214
Восполнение отсутствующих данных.....	216

7.2. Преобразование данных	218
Устранение дубликатов	218
Преобразование данных с помощью функции или отображения	220
Замена значений	221
Переименование индексов осей	222
Дискретизация и группировка по интервалам	223
Обнаружение и фильтрация выбросов	226
Перестановки и случайная выборка	227
Вычисление индикаторных переменных	229
7.3. Расширение типов данных	232
7.4. Манипуляции со строками	235
Встроенные методы строковых объектов	235
Регулярные выражения	237
Строковые функции в pandas	240
7.5. Категориальные данные	243
Для чего это нужно	244
Расширенный тип Categorical в pandas	245
Вычисления с объектами Categorical	248
Категориальные методы	250
7.6. Заключение	253

Глава 8. Переформатирование данных:

соединение, комбинирование и изменение формы..... 254

8.1. Иерархическое индексирование	254
Переупорядочение и уровни сортировки	257
Сводная статистика по уровню	258
Индексирование столбцами DataFrame	258
8.2. Комбинирование и слияние наборов данных	260
Слияние объектов DataFrame как в базах данных	260
Соединение по индексу	265
Конкатенация вдоль оси	269
Комбинирование перекрывающихся данных	274
8.3. Изменение формы и поворот	276
Изменение формы с помощью иерархического индексирования	276
Поворот из «длинного» в «широкий» формат	279
Поворот из «широкого» в «длинный» формат	282
8.4. Заключение	284

Глава 9. Построение графиков и визуализация..... 285

9.1. Краткое введение в API библиотеки matplotlib	286
Рисунки и подграфики	287
Цвета, маркеры и стили линий	291
Риски, метки и надписи	292
Аннотации и рисование в подграфике	295
Сохранение графиков в файле	297

Конфигурирование matplotlib	298
9.2. Построение графиков с помощью pandas и seaborn.....	299
Линейные графики.....	299
Столбчатые диаграммы	302
Гистограммы и графики плотности	308
Диаграммы рассеяния.....	310
Фасетные сетки и категориальные данные	313
9.3. Другие средства визуализации для Python	315
9.4. Заключение	316

Глава 10. Агрегирование данных и групповые операции 317

10.1. Как представлять себе групповые операции	318
Обход групп.....	322
Выборка столбца или подмножества столбцов	323
Группировка с помощью словарей и объектов Series	324
Группировка с помощью функций	325
Группировка по уровням индекса.....	325
10.2. Агрегирование данных	326
Применение функций, зависящих от столбца, и нескольких функций	328
Возврат агрегированных данных без индексов строк	332
10.3. Метод apply: общий принцип	
разделения–применения–объединения	332
Подавление групповых ключей.....	334
Квантильный и интервальный анализы.....	335
Пример: подстановка зависящих от группы значений	
вместо отсутствующих	337
Пример: случайная выборка и перестановка	339
Пример: групповое взвешенное среднее и корреляция.....	341
Пример: групповая линейная регрессия	343
10.4. Групповые преобразования и «развернутая» группировка	343
10.5. Сводные таблицы и перекрестная табуляция	347
Перекрестная табуляция: crosstab.....	350
10.5. Заключение.....	351

Глава 11. Временные ряды 352

11.1. Типы данных и инструменты, относящиеся к дате и времени	353
Преобразование между строкой и datetime	354
11.2. Основы работы с временными рядами.....	356
Индексирование, выборка, подмножества	358
Временные ряды с неуникальными индексами.....	360
11.3. Диапазоны дат, частоты и сдвиг	361
Генерирование диапазонов дат	362
Частоты и смещения дат	364
Сдвиг данных (с опережением и с запаздыванием)	366
11.4. Часовые пояса.....	369
Локализация и преобразование	369

Операции над объектами Timestamp с учетом часового пояса	371
Операции над датами из разных часовых поясов	372
11.5. Периоды и арифметика периодов	373
Преобразование частоты периода	374
Квартальная частота периода	376
Преобразование временных меток в периоды и обратно	377
Создание PeriodIndex из массивов	379
11.6. Передискретизация и преобразование частоты	380
Понижающая передискретизация	382
Повышающая передискретизация и интерполяция	384
Передискретизация периодов	386
Групповая передискретизация по времени	387
11.7. Скользящие оконные функции	389
Экспоненциально взвешенные функции	392
Бинарные скользящие оконные функции	394
Скользящие оконные функции, определенные пользователем	395
11.8. Заключение	396

Глава 12. Введение в библиотеки моделирования на Python..... 397

12.1. Интерфейс между pandas и кодом модели	397
12.2. Описание моделей с помощью Patsy	400
Преобразование данных в формулах Patsy	402
Категориальные данные и Patsy	404
12.3. Введение в statsmodels	406
Оценивание линейных моделей	407
Оценивание процессов с временными рядами	409
12.4. Введение в scikit-learn	410
12.5. Заключение	414

Глава 13. Примеры анализа данных..... 415

13.1. Набор данных Bitly с сайта 1.usa.gov	415
Подсчет часовых поясов на чистом Python	416
Подсчет часовых поясов с помощью pandas	418
13.2. Набор данных MovieLens 1M	424
Измерение несогласия в оценках	428
13.3. Имена, которые давали детям в США за период с 1880 по 2010 год....	432
Анализ тенденций в выборе имен	437
13.4. База данных о продуктах питания министерства сельского хозяйства США	446
13.5. База данных Федеральной избирательной комиссии	451
Статистика пожертвований по роду занятий и месту работы	454
Распределение суммы пожертвований по интервалам	457
Статистика пожертвований по штатам	459
13.6. Заключение	460

Приложение А. Дополнительные сведения**о библиотеке NumPy 461**

А.1. Внутреннее устройство объекта ndarray.....	461
Иерархия типов данных в NumPy	462
А.2. Дополнительные манипуляции с массивами.....	463
Изменение формы массива	464
Упорядочение элементов массива в С и в Fortran.....	465
Конкатенация и разбиение массива	466
Эквиваленты прихотливого индексирования: функции take и put	470
А.3. Укладывание	471
Укладывание по другим осям.....	474
Установка элементов массива с помощью укладывания	476
А.4. Дополнительные способы использования универсальных функций.....	477
Методы экземпляра u-функций	477
Написание новых u-функций на Python.....	479
А.5. Структурные массивы и массивы записей	480
Вложенные типы данных и многомерные поля.....	481
Зачем нужны структурные массивы?.....	482
А.6. Еще о сортировке.....	482
Косвенная сортировка: методы argsort и lexsort.....	483
Альтернативные алгоритмы сортировки	485
Частичная сортировка массивов	485
Метод numpy.searchsorted: поиск элементов в отсортированном массиве	486
А.7. Написание быстрых функций для NumPy с помощью Numba	487
Создание пользовательских объектов numpy.ufunc с помощью Numba	489
А.8. Дополнительные сведения о вводе-выводе массивов.....	489
Файлы, отображенные на память.....	489
HDF5 и другие варианты хранения массива.....	491
А.9. Замечания о производительности	491
Важность непрерывной памяти	491

Приложение В. Еще о системе IPython..... 494

В.1. Комбинации клавиш	494
В.2. О магических командах	495
Команда %run	497
Исполнение кода из буфера обмена.....	498
В.3. История команд	499
Поиск в истории команд и повторное выполнение.....	500
Входные и выходные переменные	500
В.4. Взаимодействие с операционной системой.....	501
Команды оболочки и псевдонимы	502
Система закладок на каталоги	503

В.5. Средства разработки программ.....	504
Интерактивный отладчик.....	504
Хронометраж программы: %time и %timeit	508
Простейшее профилирование: %prun и %run -p.....	510
Построчное профилирование функции.....	512
В.6. Советы по продуктивной разработке кода с использованием IPython	514
Перезагрузка зависимостей модуля.....	514
Советы по проектированию программ.....	515
В.7. Дополнительные возможности IPython	516
Профили и конфигурирование.....	516
В.8. Заключение	517
Предметный указатель	518

Об авторе

Уэс Маккинни – разработчик программного обеспечения и предприниматель из Нэшвилла. Получив степень бакалавра математики в МТИ в 2007 году, он поступил на работу в компанию AQR Capital Management в Гринвиче, где занимался финансовой математикой. Неудовлетворенный малоприспособными средствами анализа данных, Уэс изучил язык Python и приступил к созданию того, что в будущем стало проектом pandas. Сейчас он активный член сообщества обработки данных на Python и агитирует за использование Python в анализе данных, финансовых задачах и математической статистике.

Впоследствии Уэс стал сооснователем и генеральным директором компании DataPad, технологические активы и коллектив которой в 2014 году приобрела компания Cloudera. С тех пор он занимается технологиями больших данных и является членом комитетов по управлению проектами Apache Arrow и Apache Parquet, курируемых фондом Apache Software Foundation. В 2018 году он основал компанию Ursa Labs, некоммерческую организацию, ориентированную на разработку проекта Apache Arrow в партнерстве с компаниями RStudio и Two Sigma Investments. В 2021 году вошел в число учредителей технологического стартапа Voltron Data, где в настоящее время занимает пост технического директора.

Об иллюстрации на обложке

На обложке книги изображена перохвостая тупайя (*Ptilocercus lowii*). Это единственный представитель своего вида в семействе *Ptilocercidae* рода *Ptilocercus*; остальные тупайи принадлежат семейству *Tupaiaidae*. Тупайи отличаются длинным хвостом и мягким буро-желтым мехом. У перохвостой тупайи хвост напоминает птичье перо, за что она и получила свое название. Тупайи всеядны, питаются преимущественно насекомыми, фруктами, семенами и небольшими позвоночными животными.

Эти дикие млекопитающие, обитающие в основном в Индонезии, Малайзии и Таиланде, известны хроническим потреблением алкоголя. Как выяснилось, малайские тупайи несколько часов в сутки пьют естественно ферментированный нектар пальмы *Eugeissona tristis*, что эквивалентно употреблению от 10 до 12 стаканов вина, содержащего 3.8 % алкоголя. Но это не приводит к интоксикации благодаря развитой способности расщеплять этиловый спирт, включая его в обмен веществ способами, недоступными человеку. Кроме того, поражает отношение массы мозга к массе тела – оно больше, чем у всех прочих млекопитающих, включая и человека.

Несмотря на название, перохвостая тупайя не является настоящей тупайей, а ближе к приматам. Вследствие такого родства перохвостые тупайи стали альтернативой приматам в медицинских экспериментах по изучению миопии, психосоциального стресса и гепатита.

Предисловие от издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в основном тексте или программном коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу dmkpress@gmail.com, и мы исправим это в следующих тиражах.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательство «ДМК Пресс» очень серьезно относится к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Предисловие

Первое издание этой книги вышло в 2012 году, когда Python-библиотеки для анализа данных с открытым исходным кодом (в частности, pandas) были еще вновь и быстро развивались. Когда в 2016–2017 годах пришло время написать второе издание, мне пришлось не только привести текст в соответствие с версией Python (в первом издании использовалась версия Python 2.7), но и учесть многочисленные изменения, внесенные в pandas за прошедшие пять лет. Теперь, в 2022 году, изменений в Python оказалось меньше (сейчас текущей является версия 3.10, а на подходе 3.11), зато развитие pandas продолжилось.

В третьем издании я ставил целью привести материал в соответствие с текущими версиями Python, NumPy, pandas и другими проектами, не слишком увлекаясь обсуждением новых проектов на Python, появившихся за последние пять лет. Поскольку книга стала важным ресурсом для многих университетских курсов и практикующих профессионалов, я постараюсь избегать тем, которые рискуют устареть через год-другой. Это позволит сохранить актуальность печатных экземпляров в 2023–2024 годах и, возможно, в более длительной перспективе.

В третьем издании добавилось новшество – открытый доступ к онлайн-версии, размещенной на моем сайте <https://wesmckinney.com/book>. Это будет удобно для владельцев печатных и цифровых версий книги. Я собираюсь поддерживать текст в более-менее актуальном состоянии, поэтому если вы встретите в печатной версии что-то не работающее, как должно, всегда можно будет справиться с последними обновлениями.

ГРАФИЧЕСКИЕ ВЫДЕЛЕНИЯ

В книге применяются следующие графические выделения.

Курсив

Новые термины, URL-адреса, адреса электронной почты, имена и расширения имен файлов.

Моноширинный

Листинги программ, а также элементы кода в основном тексте: имена переменных и функций, базы данных, типы данных, переменные окружения, предложения и ключевые слова языка.

Моноширинный полужирный

Команды или иной текст, который должен быть введен пользователем буквально.

Моноширинный курсив

Текст, вместо которого следует подставить значения, заданные пользователем или определяемые контекстом.



Так обозначается совет или рекомендация.



Так обозначается замечание общего характера.



Так обозначается предупреждение или предостережение.

О ПРИМЕРАХ КОДА

Файлы данных и прочие материалы, организованные по главам, можно найти в репозитории книги на GitHub по адресу <http://github.com/wesm/pydata-book> или на его зеркале на Gitee (для тех, у кого нет доступа к GitHub) по адресу <https://gitee.com/wesmckinn/pydata-book>.

Эта книга призвана помогать вам в работе. Поэтому вы можете использовать приведенный в ней код в собственных программах и в документации. Спрашивать у нас разрешение необязательно, если только вы не собираетесь воспроизводить значительную часть кода. Например, никто не возбраняет включить в свою программу несколько фрагментов кода из книги. Однако для продажи или распространения примеров из книг издательства O'Reilly на компакт-диске разрешение требуется. Цитировать книгу и примеры в ответах на вопросы можно без ограничений. Но для включения значительных объемов кода в документацию по собственному продукту нужно получить разрешение.

Мы высоко ценим, хотя и не требуем, ссылки на наши издания. В ссылке обычно указываются название книги, имя автора, издательство и ISBN, например: «Python for Data Analysis by Wes McKinney (O'Reilly). Copyright 2022 Wes McKinney, 78-1-098-10403-0».

Если вы полагаете, что планируемое использование кода выходит за рамки изложенной выше лицензии, пожалуйста, обратитесь к нам по адресу permissions@oreilly.com.

КАК С НАМИ СВЯЗАТЬСЯ

Вопросы и замечания по поводу этой книги отправляйте в издательство:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

800-998-9938 (в США и Канаде)
707-829-0515 (международный или местный)
707-829-0104 (факс)

Для этой книги имеется веб-страница, на которой публикуются списки замеченных ошибок, примеры и прочая дополнительная информация. Адрес страницы: http://bit.ly/python_data_analysis_3e.

Замечания и вопросы технического характера следует отправлять по адресу bookquestions@oreilly.com.

Дополнительную информацию о наших книгах, конференциях и новостях вы можете найти на нашем сайте по адресу <http://www.oreilly.com>.

Ищите нас в LinkedIn: <https://linkedin.com/company/oreilly-media>.

Следите за нашей лентой в Twitter: <http://twitter.com/oreillymedia>.

Смотрите нас на YouTube: <http://www.youtube.com/oreillymedia>.

Благодарности

Эта книга – плод многолетних плодотворных обсуждений и совместной работы с многочисленными людьми со всего света. Хочу поблагодарить некоторых из них.

Памяти Джона Д. Хантера (1968–2012)

28 августа 2012 года после многолетней борьбы с раком толстой кишки ушел из жизни наш дорогой друг и коллега Джон Д. Хантер. Это произошло почти сразу после того, как я закончил рукопись первого издания книги.

Роль и влияние Джона на сообщества, специализирующиеся на применении Python в научных приложениях и обработке данных, трудно переоценить. Помимо разработки библиотеки `matplotlib` в начале 2000-х годов (время, когда Python был далеко не так популярен, как сейчас), он помогал формировать культуру целого поколения разработчиков открытого кода, ставших впоследствии столпами экосистемы Python, которую мы часто считаем самой собой разумеющейся.

Мне повезло познакомиться с Джоном в начале своей работы над открытым кодом в январе 2010-го, сразу после выхода версии `pandas` 0.1. Его вдохновляющее руководство помогало мне даже в самые тяжелые моменты не отказываться от своего видения `pandas` и Python как полноправного языка для анализа данных.

Джон был очень близок с Фернандо Пересом (Fernando Perez) и Брайаном Грейнджером (Brian Granger), заложившими основы IPython, Jupyter и выступавшими авторами многих других инициатив в сообществе Python. Мы надеялись работать над книгой вчетвером, но в итоге только у меня оказалось достаточно свободного времени. Я уверен, что он гордился бы тем, чего мы достигли, порознь и совместно, за прошедшие девять лет.

Благодарности к третьему изданию (2022)

Прошло уже больше десяти лет с тех пор, как я начал работать над первым изданием этой книги, и больше пятнадцати с момента, когда я начал карьеру

программиста на Python. За это время многое изменилось! Из нишевого языка для анализа данных Python превратился в самый популярный и распространенный язык, лежащий в основе значительной (если не подавляющей!) части науки о данных, машинного обучения и искусственного интеллекта.

Я не принимал активного участия в развитии проекта с открытым исходным кодом pandas с 2013 года, но сложившееся вокруг него всемирное сообщество разработчиков процветало и служило образцом разработки открытого ПО силами сообщества. Многие Python-проекты «следующего поколения» для работы с табличными данными строят пользовательские интерфейсы по образцу pandas, так что проект продолжает оказывать неослабевающее влияние на пути развития экосистемы науки о данных на Python.

Я надеюсь, что эта книга остается ценным подспорьем для студентов и всех, кто хочет узнать о работе с данными на Python.

Я очень признателен издательству O'Reilly, разрешившему мне опубликовать на моем сайте <https://wesmckinney.com/book> версию книги с «открытым доступом», которая, как я надеюсь, откроет мир анализа для еще большего количества людей. Дж. Дж. Аллэр (J. J. Allaire) оказался той палочкой-выручалочкой, благодаря которой мне удалось перенести текст книги из Docbook XML в Quarto (<https://quarto.org/>), замечательную новую систему подготовки научно-технических текстов для печати и публикации в вебе.

Отдельное спасибо техническим рецензентам Полу Бэрри (Paul Barry), Жану-Кристофу Лейдеру (Jean-Christophe Leyder), Абдулле Карасану (Abdullah Karasan) и Уильяму Джамиру (William Jamir) за их подробные отзывы, значительно улучшившие книгу с точки зрения удобочитаемости, ясности и понятности содержимого.

Благодарности ко второму изданию (2017)

Прошло почти пять лет с момента, когда я закончил рукопись первого издания книги в июле 2012 года. С тех пор многое изменилось. Сообщество Python неизмеримо выросло, а сложившаяся вокруг него экосистема программных продуктов с открытым исходным кодом процветает.

Новое издание не появилось бы на свет без неустанных усилий разработчиков ядра pandas, благодаря которым этот проект и сложившееся вокруг него сообщество превратились в один из краеугольных камней экосистемы Python в области науки о данных. Назову лишь некоторых: Том Аугспургер (Tom Augspurger), Йорис ван ден Боше (Joris van den Bossche), Крис Бартак (Chris Bartak), Филлип Клауд (Phillip Cloud), gflyoung, Энди Хэйдэн (Andy Hayden), Масааки Хорикоши (Masaaki Horikoshi), Стивен Хойер (Stephan Hoyer), Адам Клейн (Adam Klein), Вouter Овермейер (Wouter Overmeire), Джэфф Ребэк (Jeff Reback), Чань Ши (Chang She), Скиппер Сиболд (Skipper Seabold), Джефф Трэтнер (Jeff Tratner) и у-р.

Что касается собственно подготовки издания, я благодарю сотрудников издательства O'Reilly, которые терпеливо помогали мне на протяжении всего процесса работы над книгой, а именно: Мари Божуро (Marie Beaugureau), Бена Лорика (Ben Lorica) и Коллин Топорек (Colleen Toporek). В очередной раз у меня были замечательные технические редакторы: Том Аугспургер, Пол Бэрри (Paul Barry), Хью Браун (Hugh Brown), Джонатан Коу (Jonathan Coe) и Андреас Муллер (Andreas Muller). Спасибо вам.

Первое издание книги переведено на ряд иностранных языков, включая китайский, французский, немецкий, японский, корейский и русский. Перевод этого текста с целью сделать его доступным более широкой аудитории – трудное и зачастую неблагодарное занятие. Благодарю вас за то, что вы помогаете людям во всем мире учиться программировать и использовать средства анализа данных.

Мне также повезло пользоваться на протяжении нескольких последних лет поддержкой своих трудов по разработке ПО с открытым исходным кодом со стороны сайта Cloudera и фонда Two Sigma Investments. В то время как открытые проекты получают все меньший объем ресурсов, несопоставимый с количеством пользователей, очень важно, чтобы коммерческие компании поддерживали разработку ключевых программных проектов. Это было бы правильно.

Благодарности к первому изданию (2012)

Мне было бы трудно написать эту книгу без поддержки со стороны многих людей.

Из сотрудников издательства O'Reilly я крайне признателен редакторам Меган Бланшетт (Meghan Blanchette) и Джулии Стил (Julie Steele), которые направляли меня на протяжении всего процесса. Майк Лоукидес (Mike Loukides) также работал со мной на стадии подачи предложения и помогал с выпуском книги в свет.

В техническом рецензировании книги принимало участие много народу. Мартин Лас (Martin Blais) и Хью Браун (Hugh Brown) оказали неоценимую помощь в повышении качества примеров, ясности изложения и улучшении организации книги в целом. Джеймс Лонг (James Long), Дрю Конвей (Drew Conway), Фернандо Перес, Брайан Грейнджер, Томас Клуйвер (Thomas Kluyver), Адам Клейн, Джон Клейн, Чань Ши и Стефан ван дер Вальт (Stefan van der Walt) отрецензировали по одной или по нескольким главам и предложили ценные замечания с разных точек зрения.

Я почерпнул немало отличных идей для примеров и наборов данных в беседах с друзьями и коллегами, в том числе Майком Дьюаром (Mike Dewar), Джеффом Хаммербахером (Jeff Hammerbacher), Джеймсом Джондроу (James Johnndrow), Кристианом Ламом (Kristian Lum), Адамом Клейном, Хилари Мейсон (Hilary Mason), Чань Ши и Эшли Вильямсом (Ashley Williams).

Конечно, я в долгу перед многими лидерами сообщества, занимающегося применением открытого ПО на Python в научных приложениях, поскольку именно они заложили фундамент моей работы и воодушевляли меня, пока я писал книгу. Это люди, разрабатывающие ядро IPython (Фернандо Перес, Брайан Грейнджер, Мин Рэган-Келли, Томас Клуйвер и другие), Джон Хантер, Скиппер Сиболд, Трэвис Олифант (Travis Oliphant), Питер Вонг (Peter Wang), Эрик Джонс (Eric Jones), Роберт Керн (Robert Kern), Джозеф Перктольд (Josef Perktold), Франческ Альтед (Francesc Alted), Крис Фоннесбек (Chris Fonnesbeck) и многие, многие другие. Еще несколько человек оказывали мне значительную поддержку, делились идеями и подбадривали на протяжении всего пути: Дрю Конвей, Шон Тэйлор (Sean Taylor), Джузеппе Палеолого (Giuseppe Paleologo), Джаред Дандер (Jared Lander), Дэвид Эпштейн (David Epstein), Джон Кроуос (John Krowas), Джошуа Блум (Joshua Bloom), Дэн Пилсуорт (Den Pilsworth), Джон Майлз-Уайт (John Myles-White) и многие другие, о которых я забыл.

Я также благодарен многим, кто оказал влияние на мое становление как ученого. В первую очередь это мои бывшие коллеги по компании AQR, которые поддерживали мою работу над *pandas* в течение многих лет: Алекс Рейфман (Alex Reyfman), Майкл Вонг (Michael Wong), Тим Сарджен (Tim Sargen), Октай Курбанов (Oktay Kurbanov), Мэтью Щанц (Matthew Tschantz), Рони Израэлов (Roni Israelov), Майкл Кац (Michael Katz), Крис Уга (Chris Uga), Прасад Раманан (Prasad Ramanan), Тэд Сквэр (Ted Square) и Хун Ким (Hoon Kim). И наконец, благодарю моих университетских наставников Хэйнса Миллера (МТИ) и Майка Уэста (университет Дьюк).

Если говорить о личной жизни, то я благодарен Кэйси Динкин (Casey Dinkin), чью каждодневную поддержку невозможно переоценить, ту, которая терпела перепады моего настроения, когда я пытался собрать окончательный вариант рукописи в дополнение к своему и так уже перегруженному графику. А также моим родителям, Биллу и Ким, которые учили меня никогда не отступать от мечты и не соглашаться на меньшее.

Предварительные сведения

1.1. О чем эта книга?

Эта книга посвящена вопросам преобразования, обработки, очистки данных и вычислениям на языке Python. Моя цель – предложить руководство по тем частям языка программирования Python и экосистемы его библиотек и инструментов, относящихся к обработке данных, которые помогут вам стать хорошим аналитиком данных. Хотя в названии книги фигурируют слова «анализ данных», основной упор сделан на программировании на Python, библиотеках и инструментах, а не на методологии анализа данных как таковой. Речь идет о программировании на Python, необходимом для анализа данных.

Спустя некоторое время после выхода этой книги в 2012 году термин «наука о данных» (data science) стали употреблять для всего на свете: от простой описательной статистики до более сложного статистического анализа и машинного обучения. Открытая экосистема для анализа данных (или науки о данных) на Python с тех пор также значительно расширилась. Сейчас имеется много книг, посвященных специально более продвинутым методологиям. Лыщу себя надеждой, что эта книга подготовит вас к изучению ресурсов, ориентированных на конкретные области применения.



Возможно, кто-то сочтет, что книга в большей степени посвящена «манипулированию данными», а не «анализу данных». Мы используем также термины «первичная обработка данных» (data wrangling) и «подготовка данных» (data munging) в качестве синонимов «манипулированию данными».

Какого рода данные?

Говоря «данные», я имею в виду прежде всего *структурированные данные*; это намеренно расплывчатый термин, охватывающий различные часто встречающиеся виды данных, как то:

- табличные данные, когда данные в разных столбцах могут иметь разный тип (строки, числа, даты или еще что-то). Сюда относятся данные, которые обычно хранятся в реляционных базах или в файлах с запятой в качестве разделителя;
- многомерные списки (матрицы);

- данные, представленные в виде нескольких таблиц, связанных между собой по ключевым столбцам (то, что в SQL называется первичными и внешними ключами);
- равноотстоящие и неравноотстоящие временные ряды.

Этот список далеко не полный. Значительную часть наборов данных можно преобразовать к структурированному виду, более подходящему для анализа и моделирования, хотя сразу не всегда очевидно, как это сделать. В тех случаях, когда это не удастся, иногда есть возможность извлечь из набора данных структурированное множество признаков. Например, подборку новостных статей можно преобразовать в таблицу частот слов, к которой затем применить анализ эмоциональной окраски.

Большинству пользователей электронных таблиц типа Microsoft Excel, пожалуй, самого широко распространенного средства анализа данных, такие виды данных хорошо знакомы.

1.2. Почему именно Python?

Для многих людей (и меня в том числе) Python – язык, в который нельзя не влюбиться. С момента своего появления в 1991 году Python стал одним из самых популярных динамических языков программирования наряду с Perl, Ruby и другими. Относительно недавно Python и Ruby приобрели особую популярность как средства создания веб-сайтов в многочисленных каркасах, например Rails (Ruby) и Django (Python). Такие языки часто называют *скриптовыми*, потому что они используются для быстрого написания небольших программ – *скриптов*. Лично мне термин «скриптовый язык» не нравится, поскольку он наводит на мысль, будто для создания ответственного программного обеспечения язык не годится. Из всех интерпретируемых языков Python выделяется большим и активным сообществом научных расчетов и анализа данных. За последние 20 лет Python перешел из разряда ультрасовременного языка научных расчетов, которым пользуются на свой страх и риск, в один из самых важных языков, применяемых в науке о данных, машинном обучении и разработке ПО общего назначения в академических учреждениях и промышленности.

В области анализа данных и интерактивных научно-исследовательских расчетов с визуализацией результатов Python неизбежно приходится сравнивать со многими предметно-ориентированными языками программирования и инструментами – с открытым исходным кодом и коммерческими, такими как R, MATLAB, SAS, Stata и другими. Сравнительно недавнее появление улучшенных библиотек для Python (прежде всего pandas) сделало его серьезным конкурентом в решении задач манипулирования данными. В сочетании с достоинствами Python как универсального языка программирования это делает его отличным выбором для создания приложений обработки данных.

Python как клей

Своим успехом в области научных расчетов Python отчасти обязан простоте интеграции с кодом на C, C++ и FORTRAN. Во многих современных вычислительных средах применяется общий набор унаследованных библиотек, напи-

санных на FORTRAN и C, содержащих реализации алгоритмов линейной алгебры, оптимизации, интегрирования, быстрого преобразования Фурье и других. Поэтому многочисленные компании и национальные лаборатории используют Python как «клей» для объединения написанных за много лет программ.

Большинство программ содержат небольшие участки кода, на выполнение которых уходит большая часть времени, и большие куски «склеивающего кода», который выполняется нечасто. Во многих случаях время выполнения склеивающего кода несущественно, реальную отдачу дает оптимизация узких мест, которые иногда имеет смысл переписать на низкоуровневом языке типа C.

Решение проблемы «двух языков»

Во многих организациях принято для научных исследований, создания опытных образцов и проверки новых идей использовать предметно-ориентированные языки типа MATLAB или R, а затем переносить удачные разработки в производственную систему, написанную на Java, C# или C++. Но все чаще люди приходят к выводу, что Python подходит не только для стадий исследования и создания прототипа, но и для построения самих производственных систем. Я полагаю, что компании все чаще будут выбирать этот путь, потому что использование одного и того же набора программных средств учеными и технологами несет несомненные выгоды организации.

За последнее десятилетие появились новые подходы к решению проблемы «двух языков», в частности язык программирования Julia. Во многих случаях для получения максимальной пользы от Python *необходимо* программировать на языке более низкого уровня типа C или C++ и создавать интерфейс между таким кодом и Python. Вместе с тем технология «своевременных» (JIT) компиляторов, предлагаемая такими библиотеками, как Numba, позволила добиваться великолепной производительности многих численных алгоритмов, не покидая среду программирования на Python.

Недостатки Python

Python – великолепная среда для создания приложений для научных расчетов и большинства систем общего назначения, но тем не менее существуют задачи, для которых Python не очень подходит.

Поскольку Python – интерпретируемый язык программирования, в общем случае написанный на нем код работает значительно медленнее, чем эквивалентный код на компилируемом языке типа Java или C++. Но поскольку *время программиста* обычно стоит гораздо дороже *времени процессора*, многих такой компромисс устраивает. Однако в приложениях, где задержка должна быть очень мала (например, в торговых системах с большим количеством транзакций), время, потраченное на программирование на низкоуровневом и не обеспечивающем максимальную продуктивность языке типа C++, во имя достижения максимальной производительности, будет потрачено не зря.

Python – не идеальный язык для программирования многопоточных приложений с высокой степенью параллелизма, особенно при наличии многих потоков, активно использующих процессор. Проблема связана с наличием *глобальной блокировки интерпретатора* (GIL) – механизма, который не дает