



Содержание

| | |
|--|----|
| Предисловие | 14 |
| Об авторе | 20 |
| Об иллюстрации на обложке | 21 |
| 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 | 26 |
| matplotlib | 27 |
| IPython и Jupyter | 27 |
| SciPy | 28 |
| scikit-learn | 28 |
| statsmodels | 29 |
| 1.4. Установка и настройка | 30 |
| Windows | 30 |
| Apple OS X | 30 |
| GNU/Linux | 31 |
| Установка или обновление Python-пакетов | 31 |

| | |
|---|-----------|
| Python 2 и Python 3 | 32 |
| Интегрированные среды разработки (IDE)..... | 32 |
| 1.5. Сообщество и конференции | 33 |
| 1.6. Структура книги..... | 34 |
| Примеры кода | 34 |
| Данные для примеров | 35 |
| Соглашения об импорте..... | 35 |
| Жаргон | 35 |
| Глава 2. Основы языка Python, IPython и Jupyter-блокноты | 36 |
| 2.1. Интерпретатор Python | 37 |
| 2.2. Основы IPython | 38 |
| Запуск оболочки IPython..... | 38 |
| Запуск Jupyter-блокнота | 39 |
| Завершение по нажатию клавиши Tab | 42 |
| Интрорспекция..... | 43 |
| Команда %run..... | 45 |
| Исполнение кода из буфера обмена..... | 46 |
| Комбинации клавиш | 47 |
| О магических командах | 48 |
| Интеграция с matplotlib..... | 50 |
| 2.3. Основы языка Python | 51 |
| Семантика языка..... | 51 |
| Скалярные типы | 59 |
| Поток управления..... | 66 |
| Глава 3. Встроенные структуры данных, функции и файлы | 71 |
| 3.1. Структуры данных и последовательности | 71 |
| Кортеж..... | 71 |
| Список..... | 74 |
| Встроенные функции последовательностей | 79 |
| Словарь..... | 81 |
| Множество..... | 85 |
| Списковое, словарное и множественное включения | 87 |
| 3.2. Функции | 89 |
| Пространства имен, области видимости и локальные функции | 90 |
| Возврат нескольких значений | 91 |
| Функции являются объектами..... | 91 |
| Анонимные (лямбда) функции..... | 93 |
| Каррирование: фиксирование части аргументов..... | 94 |
| Генераторы | 94 |
| Обработка исключений..... | 97 |

| | | |
|-----------------|---|------------|
| 3.3. | Файлы и операционная система | 100 |
| | Байты и Unicode в применении к файлам | 102 |
| 3.4. | Заключение | 104 |
| Глава 4. | Основы NumPy: массивы и векторные вычисления | 105 |
| 4.1. | NumPy ndarray: объект многомерного массива | 107 |
| | Создание ndarray | 108 |
| | Тип данных для ndarray | 110 |
| | Арифметические операции с массивами NumPy | 113 |
| | Индексирование и вырезание | 114 |
| | Булево индексирование | 119 |
| | Прихотливое индексирование | 121 |
| | Транспонирование массивов и перестановка осей | 123 |
| 4.2. | Универсальные функции: быстрые поэлементные операции над массивами | 125 |
| 4.3. | Программирование с применением массивов | 127 |
| | Запись логических условий в виде операций с массивами | 129 |
| | Математические и статистические операции | 131 |
| | Методы булевых массивов | 132 |
| | Сортировка | 133 |
| | Устранение дубликатов и другие теоретико-множественные операции | 134 |
| 4.4. | Файловый ввод-вывод массивов | 135 |
| 4.5. | Линейная алгебра | 136 |
| 4.6. | Генерация псевдослучайных чисел | 138 |
| 4.7. | Пример: случайное блуждание | 139 |
| | Моделирование сразу нескольких случайных блужданий | 141 |
| 4.8. | Заключение | 142 |
| Глава 5. | Первое знакомство с pandas | 143 |
| 5.1. | Введение в структуры данных pandas | 144 |
| | Объект Series | 144 |
| | Объект DataFrame | 148 |
| | Индексные объекты | 154 |
| 5.2. | Базовая функциональность | 156 |
| | Переиндексация | 156 |
| | Удаление элементов из оси | 159 |
| | Доступ по индексу, выборка и фильтрация | 161 |
| | Целочисленные индексы | 165 |
| | Арифметические операции и выравнивание данных | 166 |
| | Применение функций и отображение | 172 |
| | Сортировка и ранжирование | 174 |
| | Индексы по осям с повторяющимися значениями | 177 |

| | | |
|------|--|-----|
| 5.3. | Редукция и вычисление описательных статистик..... | 179 |
| | Корреляция и ковариация..... | 181 |
| | Уникальные значения, счетчики значений и членство..... | 183 |
| 5.4. | Заключение | 186 |

Глава 6. Чтение и запись данных, форматы файлов..... 187

| | | |
|------|--|-----|
| 6.1. | Чтение и запись данных в текстовом формате | 187 |
| | Чтение текстовых файлов порциями..... | 193 |
| | Вывод данных в текстовом формате | 195 |
| | Обработка данных в формате с разделителями | 196 |
| | Данные в формате JSON..... | 198 |
| | XML и HTML: разбор веб-страниц..... | 200 |
| 6.2. | Двоичные форматы данных..... | 203 |
| | Формат HDF5..... | 204 |
| | Чтение файлов Microsoft Excel..... | 206 |
| 6.3. | Взаимодействие с HTML и Web API..... | 207 |
| 6.4. | Взаимодействие с базами данных..... | 209 |
| 6.5. | Заключение | 210 |

Глава 7. Очистка и подготовка данных..... 211

| | | |
|------|--|-----|
| 7.1. | Обработка отсутствующих данных | 211 |
| | Фильтрация отсутствующих данных | 213 |
| | Восполнение отсутствующих данных..... | 215 |
| 7.2. | Преобразование данных..... | 217 |
| | Устранение дубликатов..... | 217 |
| | Преобразование данных с помощью функции или отображения..... | 219 |
| | Замена значений..... | 221 |
| | Переименование индексов осей..... | 222 |
| | Дискретизация и раскладывание..... | 223 |
| | Обнаружение и фильтрация выбросов | 226 |
| | Перестановки и случайная выборка..... | 228 |
| | Вычисление индикаторных переменных..... | 229 |
| 7.3. | Манипуляции со строками | 232 |
| | Методы строковых объектов..... | 232 |
| | Регулярные выражения..... | 234 |
| | Векторные строковые функции в pandas | 237 |
| 7.4. | Заключение | 240 |

Глава 8. Переформатирование данных: соединение, комбинирование и изменение формы..... 241

| | | |
|------|---|-----|
| 8.1. | Иерархическое индексирование | 241 |
| | Переупорядочение и уровни сортировки..... | 244 |

| | |
|---|------------|
| Сводная статистика по уровню | 245 |
| Индексирование с помощью столбцов DataFrame | 246 |
| 8.2. Комбинирование и слияние наборов данных | 247 |
| Слияние объектов DataFrame как в базах данных | 247 |
| Соединение по индексу | 252 |
| Конкатенация вдоль оси | 256 |
| Комбинирование перекрывающихся данных | 261 |
| 8.3. Изменение формы и поворот | 263 |
| Изменение формы с помощью иерархического индексирования | 263 |
| Поворот из «длинного» в «широкий» формат | 266 |
| Поворот из «широкого» в «длинный» формат | 270 |
| 8.4. Заключение | 272 |
| Глава 9. Построение графиков и визуализация | 273 |
| 9.1. Краткое введение в API библиотеки matplotlib | 274 |
| Рисунки и подграфики | 275 |
| Цвета, маркеры и стили линий | 278 |
| Риски, метки и надписи | 281 |
| Аннотации и рисование в подграфике | 284 |
| Сохранение графиков в файле | 286 |
| Конфигурирование matplotlib | 288 |
| 9.2. Построение графиков с помощью pandas и seaborn | 288 |
| Линейные графики | 289 |
| Столбчатые диаграммы | 291 |
| Гистограммы и графики плотности | 296 |
| Диаграммы рассеяния | 299 |
| Фасетные сетки и категориальные данные | 301 |
| 9.3. Другие средства визуализации для Python | 303 |
| 9.4. Заключение | 303 |
| Глава 10. Агрегирование данных и групповые операции | 304 |
| 10.1. Механизм GroupBy | 305 |
| Обход групп | 308 |
| Группировка с помощью словарей и объектов Series | 311 |
| Группировка с помощью функций | 312 |
| Группировка по уровням индекса | 313 |
| 10.2. Агрегирование данных | 313 |
| Применение функций, зависящих от столбца и нескольких функций | 315 |
| Возврат агрегированных данных без индексов строк | 319 |
| 10.3. Метод apply: часть общего принципа разделения-применения-объединения | 319 |
| Подавление групповых ключей | 322 |

| | |
|---|------------|
| Квантильный и интервальный анализы..... | 322 |
| Пример: подстановка зависящих от группы значений вместо отсутствующих | 324 |
| Пример: случайная выборка и перестановка | 326 |
| Пример: групповое взвешенное среднее и корреляция | 328 |
| Пример: групповая линейная регрессия..... | 330 |
| 10.4. Сводные таблицы и перекрестное табулирование..... | 331 |
| Таблицы сопряженности..... | 334 |
| 10.5. Заключение | 335 |
| Глава 11. Временные ряды..... | 336 |
| 11.1. Типы данных и инструменты, относящиеся к дате и времени..... | 337 |
| Преобразование между строкой и datetime | 338 |
| 11.2. Основы работы с временными рядами | 341 |
| Индексирование, выборка, подмножества | 342 |
| Временные ряды с неуникальными индексами | 345 |
| 11.3. Диапазоны дат, частоты и сдвиг..... | 346 |
| Генерация диапазонов дат..... | 347 |
| Частоты и смещения дат..... | 349 |
| Сдвиг данных (с опережением и с запаздыванием)..... | 351 |
| 11.4. Часовые пояса..... | 354 |
| Локализация и преобразование..... | 355 |
| Операции над объектами Timestamp с учетом часового пояса..... | 357 |
| Операции между датами из разных часовых поясов | 358 |
| 11.5. Периоды и арифметика периодов..... | 359 |
| Преобразование частоты периода..... | 360 |
| Квартальная частота периода | 362 |
| Преобразование временных меток в периоды и обратно | 363 |
| Создание PeriodIndex из массивов..... | 365 |
| 11.6. Передискретизация и преобразование частоты..... | 367 |
| Понижающая передискретизация..... | 369 |
| Повышающая передискретизация и интерполяция..... | 371 |
| Передискретизация периодов..... | 373 |
| 11.7. Скользящие оконные функции..... | 374 |
| Экспоненциально взвешенные функции | 378 |
| Бинарные скользящие оконные функции | 379 |
| Скользящие оконные функции, определенные пользователем | 381 |
| 11.8. Заключение | 382 |
| Глава 12. Дополнительные сведения о библиотеке NumPy..... | 383 |
| 12.1. Категориальные данные..... | 383 |
| Для чего это нужно | 383 |

| | |
|---|------------|
| Категориальные типы в pandas..... | 385 |
| Вычисления с категориальными значениями..... | 388 |
| Категориальные методы..... | 390 |
| 12.2. Дополнительные способы использования GroupBy..... | 393 |
| Групповые преобразования и GroupBy с «развертыванием» | 393 |
| Групповая передискретизация по времени | 397 |
| 12.3. Сцепление методов..... | 399 |
| Метод pipe..... | 400 |
| 12.4. Заключение | 401 |
| Глава 13. Введение в библиотеки моделирования на Python..... | 402 |
| 13.1. Интерфейс между pandas и кодом модели..... | 402 |
| 13.2. Описание моделей с помощью Patsy..... | 405 |
| Преобразование данных в формулах Patsy..... | 408 |
| Категориальные данные и Patsy..... | 410 |
| 13.3. Введение в statsmodels | 412 |
| Оценивание линейных моделей | 413 |
| Оценивание процессов с временными рядами..... | 416 |
| 13.4. Введение в scikit-learn | 417 |
| 13.5. Продолжение своего образования..... | 420 |
| Глава 14. Примеры анализа данных..... | 422 |
| 14.1. 1.usa.gov data from Bitly | 422 |
| Подсчет часовых поясов на чистом Python..... | 423 |
| Подсчет часовых поясов с помощью pandas..... | 425 |
| 14.2. Набор данных MovieLens 1M | 432 |
| Измерение несогласия в оценках..... | 437 |
| 14.3. Имена, которые давали детям в США за период с 1880 по 2010 год..... | 439 |
| Анализ тенденций в выборе имен | 444 |
| 14.4. База данных о продуктах питания министерства сельского хозяйства США..... | 453 |
| 14.5. База данных федеральной избирательной комиссии | 459 |
| Статистика пожертвований по роду занятий и месту работы..... | 462 |
| Распределение суммы пожертвований по интервалам..... | 465 |
| Статистика пожертвований по штатам | 467 |
| 14.6. Заключение | 468 |
| Приложение А. Дополнительные сведения о библиотеке NumPy..... | 469 |
| A.1. Внутреннее устройство объекта ndarray..... | 469 |
| Иерархия типов данных в NumPy | 470 |
| A.2. Дополнительные манипуляции с массивами..... | 471 |

| | |
|---|------------|
| Изменение формы массива..... | 472 |
| Упорядочение элементов массива в C и в Fortran..... | 474 |
| Конкатенация и разбиение массива..... | 474 |
| Повторение элементов: функции <code>tile</code> и <code>repeat</code> | 477 |
| Эквиваленты прихотливого индексирования: функции <code>take</code> и <code>put</code> | 479 |
| A.3. Укладывание..... | 480 |
| Укладывание по другим осям..... | 482 |
| Установка элементов массива с помощью укладывания..... | 484 |
| A.4. Дополнительные способы использования универсальных функций..... | 485 |
| Методы экземпляра <code>u</code> -функций..... | 485 |
| Написание новых <code>u</code> -функций на Python..... | 488 |
| A.5. Структурные массивы..... | 489 |
| Вложенные типы данных и многомерные поля..... | 489 |
| Зачем нужны структурные массивы?..... | 490 |
| A.6. Еще о сортировке..... | 491 |
| Косвенная сортировка: методы <code>argsort</code> и <code>lexsort</code> | 492 |
| Альтернативные алгоритмы сортировки..... | 493 |
| Частичная сортировка массивов..... | 494 |
| Метод <code>numpy.searchsorted</code> : поиск элементов в отсортированном массиве..... | 495 |
| A.7. Написание быстрых функций для NumPy с помощью Numba..... | 496 |
| Создание пользовательских объектов <code>numpy.ufunc</code> с помощью Numba..... | 498 |
| A.8. Дополнительные сведения о вводе-выводе массивов..... | 498 |
| Файлы, спроецированные на память..... | 498 |
| HDF5 и другие варианты хранения массива..... | 500 |
| A.9. Замечания о производительности..... | 500 |
| Важность непрерывной памяти..... | 500 |
| Приложение В. Еще о системе IPython..... | 503 |
| V.1. История команд..... | 503 |
| Поиск в истории команд и повторное выполнение..... | 503 |
| Входные и выходные переменные..... | 504 |
| V.2. Взаимодействие с операционной системой..... | 505 |
| Команды оболочки и псевдонимы..... | 506 |
| Система закладок на каталоги..... | 507 |
| V.3. Средства разработки программ..... | 507 |
| Интерактивный отладчик..... | 507 |
| Хронометраж программы: <code>%time</code> и <code>%timeit</code> | 512 |
| Простейшее профилирование: <code>%prun</code> и <code>%run -p</code> | 514 |
| Построчное профилирование функции..... | 516 |
| V.4. Советы по продуктивной разработке кода с использованием IPython..... | 518 |
| Перезагрузка зависимостей модуля..... | 518 |

| | |
|---|------------|
| Советы по проектированию программ | 519 |
| V.5. Дополнительные возможности IPython | 521 |
| Делайте классы дружелюбными к IPython | 521 |
| Профили и конфигурирование | 521 |
| V.6. Заключение | 523 |
| Предметный указатель..... | 524 |



Предисловие

Что нового во втором издании?

Первое издание этой книги вышло в 2012 году, когда Python-библиотеки для анализа данных с открытым исходным кодом (в частности, pandas) были еще внове и быстро развивались. В этом исправленном и дополненном издании я переработал главы, стремясь отразить как несовместимые изменения и устаревшие возможности, так и новые средства, появившиеся за прошедшие пять лет. Я также добавил новый материал с описанием инструментов, которые либо еще не существовали в 2012 году, либо были недостаточно зрелыми. Наконец, я старался не писать о новых или активно разрабатываемых проектах с открытым кодом, которым, возможно, не суждено дожить до зрелости. Хотелось бы представить читателям этого издания средства, которые не утратят актуальности ни в 2020, ни в 2021 году.

Ниже перечислены основные отличия второго издания.

- Весь код, включая пособие по Python, обновлен и доведен до уровня версии 3.6 (в первом издании использовалась версия Python 2.7).
- Обновлены инструкции по установке Python из дистрибутива Anaconda Python Distribution, а также всех дополнительных Python-пакетов.
- Внесены исправления, соответствующие последним версиям библиотеки pandas, существовавшим в 2017 году.
- Добавлена глава о дополнительных средствах pandas, дающая также ряд других советов по работе с библиотекой.
- Размещено краткое введение в библиотеки statsmodels и scikit-learn.

Кроме того, материал, вошедший в первое издание, подвергнут реорганизации, чтобы книгу было проще читать начинающим пользователям.

Графические выделения

В книге применяются следующие графические выделения.

Курсив

Новые термины, 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 предостережение или предупреждение.

О примерах кода

Файлы данных и прочие материалы, организованные по главам, можно найти в репозитории книги на GitHub: <http://github.com/wesm/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 разрешения необязательно, если только вы не собираетесь воспроизводить значительную часть кода. Например, не возбраняется 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 позволение.

Мы высоко ценим (хотя и не требуем их размещения) ссылки на наши издания. В ссылке обычно указывается название книги, имя автора, издатель-

ство и ISBN, например: «Python for Data Analysis by Wes McKinney (O'Reilly). Copyright 2017 Wes McKinney, 978-1-491-95766-0».

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

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.

Скачивание исходного кода примеров

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com или www.dmk.ru на странице с описанием соответствующей книги.

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. Издательства «ДМК Пресс» и Packt очень серьезно относятся 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 электронной почты dmkpress@gmail.com.

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

Благодарности

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

Памяти Джона Д. Хантера (1968–2012)

В августе 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 само собой разумеющейся.

Мне повезло познакомиться с Джоном в начале своей работы над открытым кодом в январе 2010 года, 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), которые заложили основы IPython и Jupyter и были 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 пять лет.

Благодарности ко второму изданию (2017)

Прошло почти пять лет с того времени, как я закончил рукопись первого издания книги. Случилось это 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), gyoung, Энди Хэйдэн (Andy Hayden), Масааки Хорикоши (Masaaki Horikoshi), Стивен Хойер (Stephan Hoyer), Адам Клейн (Adam Klein), Вouter Овермейр (Wouter Overmeire), Джэфф Ребэк (Jeff Reback), Чань Ши (Chang She), Скиппер Сиболд (Skipper Seabold), Джефф Трэтнер (Jeff Tratner) и др.

Что касается собственно подготовки издания, то 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 Маллер (Andreas Muller). Спасибо вам.

Первое издание книги переведено на ряд иностранных языков, в том числе на 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 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 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 многие принимали участие. 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 Клюйвер (Thomas Kluyver), Адам

Клейн, Джон Клейн, Чань Ши и Стефан ван дер Вальт (Stefan van der Walt) отрецензировали по одной или по несколько глав и сделали 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 с Майком Дьюаром (Mike Dewar), Джеффом Хаммербахером (Jeff Hammerbacher), Джеймсом Джондроу (James Johndrow), Кристианом Ламом (Kristian Lum), Адамом Клейном, Хилари Мейсон (Hilary Mason), Чань Ши и Эшли Вильямсом (Ashley Williams).

Конечно, я в долгу у многих лидеров сообщества, применяющего открытое ПО на Python в научных приложениях, поскольку именно они заложили фундамент моей работы и воодушевляли меня, пока я писал книгу. Это те, кто разрабатывал ядро IPython (Фернандо Перес, Брайан Грейнджер, Мин Рэган-Келли, Томас Ключвер и др.), 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 всем, кто повлиял на становление меня 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 Шанц (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. Спасибо той, кто терпел перепады моего настроения, когда я пытался собрать окончательный вариант рукописи, несмотря на свой и так уже перегруженный график. Благодарю и моих родителей, Билла и Ким, которые учили меня никогда не отступать от мечты и не соглашаться на меньшее.



Об авторе

Уэс Маккини – разработчик программного обеспечения и предприниматель из Нью-Йорка. После получения степени бакалавра математики в МТИ в 2007 году его приняли на работу в компанию AQR Capital Management в Гринвиче, где занимался финансовой математикой. Неудовлетворенный малопригодными 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, курируемыми фондом Apache Software Foundation. В 2016 году автор перешел в компанию Two Sigma Investments из Нью-Йорка, где продолжает трудиться над тем, чтобы средствами ПО с открытым исходным кодом сделать анализ данных быстрее и проще.



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 10–12 стаканов вина, содержащего 3,8 % алкоголя. Но это не приводит к интоксикации их организма 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, в том числе у 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.



Глава 1. Предварительные сведения

1.1. О чем эта книга?

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, на библиотеках и инструментах, а не на методологии анализа данных как таковой. Речь идет о программировании на Python, необходимом для анализа данных.

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 – в разных столбцах они 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.

Это далеко не полный список. Значительную часть наборов данных можно привести к структурированному виду, более подходящему для анализа и моделирования, хотя сразу не всегда очевидно, как это сделать. В тех случаях, когда это не удается, 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 для меня в том числе) Python – язык, в который нельзя не влюбиться. С момента появления в 1991 году Python стал одним из самых популярных динамических языков программирования наряду с Perl, Ruby и др. 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, потому что 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 десять лет Python превратился из ультра-современного языка научных расчетов, которым пользуются на свой страх и риск, в один из самых важных языков, применяемых в науке о данных, в машинном обучении и разработке ПО общего назначения в академических учреждениях и промышленности.

При анализе 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 др. Сравнительно недавно появились улучшенные библиотеки для Python (прежде всего pandas), и он стал серьезным конкурентом в решении задач манипулирования данными. А так как Python еще – и универсальный язык программирования, то это отличный выбор 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,

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

В значительном количестве программ содержатся небольшие участки кода, на выполнение которых уходит много времени, и внушительные куски склеивающего кода, который выполняют нечасто. В большинстве случаев время выполнения склеивающего кода несущественно, реальную отдачу дает оптимизация узких мест, которые иногда имеет смысл переписать на низкоуровневом языке типа С.

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, С# или С++. Но все чаще люди приходят к выводу, что Python подходит не только для исследования и создания прототипа, но и для построения самих производственных систем. Полагаю, что компании большей частью будут выбирать этот путь, потому что использование учеными и технологами одного и того же набора программных средств, несомненно, выгодно для организации.

Недостатки Python

Python – великолепная среда для создания приложений для научных расчетов и большинства систем общего назначения, но тем не менее существуют задачи, которым 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 С++. 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 С++ 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 интерпретатору исполнять более одной команды байт-кода Python в каждый

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru