Содержание

Предисловие

Pandas – популярная библиотека Python, применяющаяся для практического анализа данных в реальных проектах. Она предлагает воспользоваться эффективными, быстрыми и высокопроизводительными структурами данных, которые упрощают предварительную обработку и анализ информации. Это учебное пособие существенно поможет вам, предоставив в ваше распоряжение внушительный набор инструментов, предлагаемых библиотекой pandas для выполнения различных операций с данными и их анализа.

О содержании книги

Глава 1 «Библиотека pandas и анализ данных» – это практическое введение в основные функции библиотеки pandas. Предназначение этой главы – дать некоторое представление об использовании библиотеки pandas в контексте статистики и науки о данных. В этой главе мы рассмотрим несколько принципов, лежащих в основе науки о данных, и покажем, как они реализованы в библиотеке pandas. Эта глава задает контекст для каждой последующей главы, связанной с наукой о данных.

Глава 2 «Запуск библиотеки pandas» проинструктирует читателя по поводу того, как загрузить и установить библиотеку pandas, и познакомит его с некоторыми базовыми понятиями библиотеки pandas. Мы также рассмотрим, как можно работать с примерами с помощью iPython и тетрадок Jupiter.

Глава 3 «Представление одномерных данных с помощью объекта Series» познакомит читателя со структурой данных Series, которая используется для представления одномерных индексированных данных. Читатель узнает о том, как создавать объекты Series и как работать с данными, хранящимися внутри этих объектов. Кроме того, он узнает об индексах и выравнивании данных, а также о том, как объект Series можно использовать для создания срезов данных.

Глава 4 «Представление табличных и многомерных данных с помощью объекта DataFrame» познакомит читателя со структурой данных DataFrame, которая используется для представления и индексации многомерных данных. В этой главе читатель научится создавать объекты DataFrame, используя различные наборы статических данных, и выполнять отбор определенных столбцов и строк внутри датафрейма. Сложные запросы, операции с данными и индексация будут рассмотрены в следующей главе.

Глава 5 «Выполнение операций над объектом DataFrame и его содержимым» расширяет предыдущую главу и расскажет о том, как выполнять более сложные операции с объектом DataFrame. Мы начнем с добавления удаления столбцов и строк, рассмотрим модификацию данных внутри объекта DataFrame (а также создание измененной копии) и выполнение арифметических операций с данными, научимся создавать иерархические индексы, а также вычислять популярные статистики по данным датафрейма.

Глава 6 «Индексация данных» расскажет об использовании различных типов индекса библиотеки pandas (Int64Index, RangeIndex, IntervalIndex, CategoricalIndex, DatetimeIndex, PeriodIndex).

Глава 7 «Категориальные данные» познакомит читателя с тем, как создавать объекты Categorical для представления категориальных данных и использовать их в работе.

В главе 8 «Численные и статистические методы» рассматриваются различные арифметические операции над объектами Series и DataFrame, а также вычисление статистик для объектов pandas.

Глава 9 «Загрузка данных» расскажет о том, как можно загрузить данные из внешних источников и записать в объекты Series и DataFrame. Кроме того, в этой главе рассматривается загрузка данных из разных источников, таких как файлы, HTTP-серверы, системы баз данных и веб-службы. Также рассматривается обработка данных в форматах CSV, HTML и JSON.

В главе 10 «Приведение данных в порядок» будет рассказано о том, как приводить данные в порядок, чтобы они были пригодны для анализа.

Глава 11 «Объединение, связывание и изменение формы данных» расскажет читателю о том, как можно взять несколько объектов pandas и объединить их с помощью операций соединения, слияния и конкатенации.

Глава 12 «Агрегация данных» расскажет о группировке и агрегации данных. В библиотеке pandas эти операции выполняются с помощью схемы «разделение – применение – объединение». Читатель научится использовать эту схему для различных способов группировки данных, а также применять агрегирующие функции для вычисления результатов по каждой группе данных.

Глава 13 «Анализ временных рядов» расскажет о том, как работать с временными рядами в библиотеке pandas. В этой главе будут освещены широкие возможности библиотеки pandas, существенно облегчающие анализ временных рядов.

Глава 14 «Визуализация» научит вас создавать визуализации данных на основе данных, хранящихся в объектах Series и DataFrame. Мы начнем с изучения основ, создания простой диаграммы настройки нескольких параметров диаграммы (настройки легенд, меток и цветов), рассмотрим создание нескольких распространенных типов графиков, которые используются для представления различных типов данных.

В приложении 1 «Советы по оптимизации вычислений в библиотеке pandas» даются некоторые рекомендации по ускорению вычислений в pandas.

Приложение 2 «Улучшение производительности pandas» представляет собой перевод одноименного раздела официального пособия по библиотеке pandas https://pandas.pydata.org/pandas-docs/stable/ enhancingperf.html.

Приложение 3 «Используем pandas для больших данных» расскажет, как за счет применения более эффективных типов данных можно уменьшить использование памяти.

В приложениях 4 и 5 на примере конкурсной задачи Tinkoff Data Science Challenge и конкурсной задачи предсказания отклика ОТП Банка детально показаны этапы предварительной обработки данных, в частности приведение переменных к нужным типам, обработка редких категорий, импутация пропусков, конструирование признаков, также освещаются специальные процедуры предварительной обработки данных, позволяющие улучшить модель логистической регрессии.

Приложение 6 «Работа с датами и строками» посвящено таким задачам, как правильный парсинг дат различного формата, изменение регистра букв в строках, удаление лишних символов из строк, извлечение нужных символов из строк.

Приложение 7 «Работа с предупреждением SettingWithCopyWarning в библиотеке pandas» посвящено причинам появления предупреждения SettingWithCopyWarning и способам его устранения.

В приложении 8 «От pandas к scikit-learn– новый подход к управлению рабочими процессами» внимание уделено работе с конвейерами.

Что необходимо для чтения этой книги

Эта книга предполагает некоторое знакомство с принципами программирования, но те, кто не имеет опыта программирования или, в частности, опыта программирования на языке Python, будут довольны примерами, поскольку они в большей степени сосредоточены на конструкциях библиотеки pandas, нежели на языке Python или программировании вообще. Примеры приводятся для Anaconda 5.2 для Python 3.6 и pandas 0.23. Если вы не установили их, в главе 2 «Запуск библиотеки pandas» дается инструкция относительно установки pandas в системах Windows, OSX и Ubuntu.

На кого рассчитана эта книга

Эта книга идеально подходит для специалистов по работе с данными, аналитиков данных и программистов Python, которые хотят погрузиться в анализ данных с использованием библиотеки pandas, и всех, кто интересуется анализом данных. Некоторые познания в области статистики и программирования помогут вам извлечь максимальную пользу из этой книги, но они не обязательны. Предварительный опыт работы с pandas также не требуется.

Соглашения

В данной книге используется несколько стилей текста для разных видов информации. Вот несколько примеров этих стилей и объяснение их смысла.

Слова в тексте, обозначающие объекты, функции, методы и другие элементы программного кода, отображаются так: «Эту информацию можно легко импортировать в датафрейм с помощью функции `pd.read_csv()` следующим образом».

Блок программного кода, введенный в интерпретаторе Python, отображается следующим образом:

```
import pandas as pd
df = pd.DataFrame.from_items ([('column1', [1, 2, 3])])
print(df)
```

Любой ввод или вывод в командной строке записывается таким образом:

```
mh@ubuntu:~/Downloads$ chmod +x Anaconda-2.1.0-Linux-x86_64.sh
mh@ubuntu:~/Downloads$ ./Anaconda-2.1.0-Linux-x86_64.sh
```

Новые термины и **важные слова** выделены жирным шрифтом.

Предупреждения и важные примечания выглядят так.

Советы и рекомендации выглядят так.

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 прямо на нашем сайте www.dmkpress.com, зайдя на страницу книги, и оставить комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com, при этом напишите название книги в теме письма.

Если есть тема, в которой вы квалифицированы, и вы 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.дмк.рф на странице с описанием соответствующей книги.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы удостовериться в качестве наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг — возможно, ошибку в тексте или в коде, — мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от расстройств и поможете нам улучшить последующие версии 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 незаконно выполненной копией любой нашей книги, пожалуйста, сообщите нам адрес копии или вебсайта, чтобы мы могли применить санкции.

Пожалуйста, свяжитесь с нами по адресу электронной почты dmkpress@gmail.com со ссылкой на подозрительные материалы.

Мы высоко ценим любую помощь по защите наших авторов, помогающую нам предоставлять вам качественные материалы.

Глава 1

Библиотека pandas и анализ данных

Добро пожаловать на страницы книги «Изучаем pandas»! В этой книге мы отправимся в путешествие, в ходе которого вы научитесь работать с pandas, библиотекой анализа данных с открытым исходным кодом, предназначенной для языка программирования Python. Библиотека pandas предлагает высокопроизводительные и простые в использовании структуры данных и инструменты анализа, созданные с помощью языка Python. Библиотека pandas привнесла в Python массу полезных инструментов, взяв их из языка статистического программирования R, в частности объекты **data frame** (**датафрейм**), пакеты R, например `plyr` и `reshape2`, и разместила их в одной библиотеке, которую вы можете использовать в среде Python.

В первой главе мы посвятим время базовому знакомству с библиотекой pandas и тому, как она вписывается в обширную картину анализа данных. Вместо того чтобы полностью сосредоточиться на конкретных аспектах использования библиотеки pandas, эта глава призвана дать читателю ощущение своего места в обширной картине анализа данных. Цель состоит в том, чтобы при изучении библиотеки pandas вы также узнали о том, зачем нужны все эти различные функции, выполняющие задачи анализа данных.

Итак, давайте начнем. В этой главе мы рассмотрим:

- что из себя представляет библиотека pandas, почему она была создана и что она вам даст;
- как библиотека pandas связана с анализом данных и наукой о данных;
- этапы анализа данных и их поддержку в библиотеке pandas;
- общие понятия «данные» и «аналитика»;
- основные понятия анализа данных и статистического анализа;
- типы данных и их использование в библиотеке pandas;
- другие библиотеки в экосистеме Python, которые вы, вероятно, будете использовать вместе с pandas.

Знакомство с библиотекой pandas

Библиотека pandas – это библиотека Python, содержащая высокоуровневые структуры данных и инструменты, которые были созданы, чтобы помочь программистам Python осуществить полноценный анализ данных. Конечная цель

библиотеки pandas заключается в том, чтобы помочь вам быстро найти необходимую информацию, скрытую в данных, при этом информацию содержательного характера.

Разработка библиотеки pandas была начата в 2008 году Уэсом Маккинни и представлена в 2009 году как проект с открытым исходным программным кодом. В настоящее время библиотека pandas активно курируется и разрабатывается различными организациями и участниками.

Первоначально библиотека pandas предназначалась для применения в финансах, в частности благодаря ее возможностям работы с временными рядами и обработке исторической информации об акциях. Обработка финансовой информации сопряжена с массой проблем, вот некоторые из них:

- обработка данных (например, данных о котировках акций), меняющихся с течением времени;
- необходимость единого стандарта измерений нескольких потоков данных в один и тот же период времени;
- определение взаимосвязи (корреляции) между двумя и более потоками данных;
- представление дат и времени в качестве объектов первого класса;
- увеличение или уменьшение шага дискретизации временного ряда.

Для выполнения этих операций необходим инструмент, который позволяет нам извлекать, индексировать, очищать и приводить в порядок, изменять и объединять данные, создавать срезы данных и выполнять различные виды анализа как для одномерных, так и для многомерных данных, включая данные разного типа, которые автоматически выравниваются по набору общих индексных меток. И вот как раз здесь на помощь приходит библиотека pandas, которая предлагает множество полезных и мощных функций, например:

- быстрые и эффективные объекты Series и DataFrame для обработки данных с помощью встроенной индексации;
- интеллектуальное выравнивание данных с помощью индексов и меток;
- интегрированная обработка пропущенных данных;
- инструменты для приведения данных в порядок;
- встроенные инструменты для чтения и записи данных для обмена между объектами Series и DataFrame в памяти, файлами, базами данных и веб-службами;
- возможность обработки данных, хранящихся в различных популярных форматах, таких как CSV, Excel, HDF5 и JSON;
- изменение формы и поворот данных;
- интеллектуальное создание срезов данных на основе меток, сложная индексация и отбор из больших наборов данных подмножеств по определенному критерию;
- удаление и вставка столбцов из объектов Series и DataFrame для изменения размера;
- агрегирование или преобразование данных с помощью мощного инструмента «разделение – применение – объединение»;
- иерархическая индексация, облегчающая работу с высокоразмерными данными в низкоразмерной структуре данных;
- высокопроизводительное слияние и соединение наборов данных;

- разнообразные функции для работы с временными рядами, включая создание диапазона дат и преобразование частоты временного ряда, вычисление скользящих статистик, скользящих линейных регрессий, смещение дат и сдвиг временного ряда с запаздыванием;
- оптимизация для достижения более высокой производительности, включающая программный код, написанный на Cython или C.

Мощный набор функций в сочетании с бесшовной интеграцией с Python и другими инструментами экосистемы Python позволил библиотеке pandas найти широкое применение во многих областях. Она используется в самых разных академических и коммерческих областях, включая финансы, нейробиологию, экономику, статистику, рекламу и веб-аналитику. Она стала одним из наиболее предпочтительных инструментов для специалистов по работе с данными.

Python долгое время широко использовался для сбора данных и подготовки, но при этом в меньшей степени был предназначен для анализа данных и моделирования. Библиотека pandas помогает заполнить этот пробел, позволяя вам выполнить весь рабочий процесс анализа данных в среде Python, не переходя на такой более специализированный язык, как R. Это очень важно, поскольку люди, знакомые с языком Python, являющимся более универсальным языком программирования, чем R (язык, ориентированный в большей степени на статистиков), получают в свое распоряжение массу функций по представлению и обработке данных, имеющихся в R, и при этом полностью остаются в невероятно богатой экосистеме Python.

В сочетании с IPython, тетрадками Jupyter и широким выбором других библиотек среда Python в плане выполнения анализа данных превосходит по производительности, эффективности и возможности совместной работы многие другие инструменты. Все это привело к тому, что многие пользователи широко применяют библиотеку pandas в самых различных отраслях.

Обработка данных, анализ, наука и библиотека pandas

Мы живем в мире, в котором каждый день генерируются и записываются огромные объемы данных. Эти данные поступают из множества информационных систем, устройств и датчиков. Почти все, что вы делаете, и все то, что вы используете в рамках своей деятельности, генерирует данные, которые можно собрать или которые уже собраны.

Во многом такая ситуация была обусловлена повсеместным распространением услуг, связанных с информационными сетями, и стремительно возросшими возможностями хранения данных. Все это в сочетании с постоянно снижающейся стоимостью хранения повысило эффективность сбора и хранения даже самых тривиальных данных.

В итоге было накоплены огромные объемы данных, готовые для загрузки. Но эти данные сосредоточились по всему киберпространству, и на самом деле их еще нельзя назвать **информацией** (**information**). Данные представляют собой коллекции зарегистрированных событий, будь то финансовые данные, или ваше общение в социальных сетях, или ваш персональный трекер здоровья, отслеживающий сердцебиение в течение дня. Эти данные хранятся в различных форматах, расположены в разных местах, и исследование сырых данных может дать ценную

информацию.

Логично, что весь процесс можно разбить на три основные дисциплины:

- обработка данных;
- анализ данных;
- наука о данных.

Эти три дисциплины часто пересекаются. Вопросы о том, где заканчивается одна дисциплина и начинается другая, остаются открытыми. В следующих разделах мы дадим определения этим дисциплинам.

Обработка данных

Данные разбросаны по всей планете. Они хранятся в разных форматах. Они имеют разный уровень качества. Поэтому существует потребность в инструментах и процессах сбора данных, дающих такое представление данных, которое можно использовать для принятия решений. Инструмент, который используется для работы с данными на этапе подготовки к анализу, должен уметь решать различные задачи. Функционал такого инструмента включает в себя:

- программируемость для повторного использования и совместного использования;
- загрузку данных из внешних источников;
- локальное сохранение данных;
- индексацию данных для их эффективного извлечения;
- выравнивание данных в разных наборах на основе атрибутов;
- объединение данных, расположенных в разных наборах;
- преобразование данных в другое представление;
- очистку данных от «мусора»;
- эффективную обработку загрязненных данных;
- группировку данных;
- агрегирование данных по схожим характеристикам;
- применение функций, вычисляющих среднее или выполняющих преобразования;
- выполнение запросов или создание срезов для исследования подмножеств данных;
- изменение формы данных;
- создание отдельных категорий данных;
- изменение частоты временного ряда.

Существует масса инструментов для обработки данных. Каждый из них отличается поддержкой элементов этого списка, способами их развертывания и способами их использования. Эти инструменты включают в себя реляционные базы данных (SQL Server, Oracle), электронные таблицы (Excel), системы обработки событий (такие как Spark) и более общие инструменты, такие как R и pandas.

Анализ данных

Анализ данных – это процесс извлечения смысла из данных. Данные, представленные в количественном виде, часто называют **информацией** (**information**). Анализ данных – это процесс получения информации из данных путем создания моделей и применения математического аппарата для поиска закономерностей.

Он часто перекликается с обработкой данных, и не всегда можно четко провести различие между ними. Многие инструменты обработки данных также содержат аналитические функции, а инструменты анализа данных нередко предлагают возможности обработки данных.

Наука о данных

Наука о данных – это процесс использования статистики и анализа данных для понимания **явлений** (**phenomena**), скрытых в данных. Наука о данных обычно начинается с информации и применяет к ней более сложный анализ на основе знаний, относящихся к разным предметным областям. К этим предметным областям относятся математика, статистика, информатика, компьютерные науки, машинное обучение, классификация, кластерный анализ, интеллектуальный анализ данных, базы данных и визуализация. Наука о данных носит многодисциплинарный характер. Ее методы анализа могут сильно отличаться друг от друга и зависеть от конкретной предметной области.

Предназначение библиотеки pandas

В первую очередь библиотека pandas – превосходный инструмент обработки данными. Все потребности, описанные ранее, будут рассмотрены в этой книге с использованием библиотеки pandas. На решение этих задач и направлен основной функционал библиотеки pandas, и именно на решении большей части таковых задач мы и сосредоточимся в этой книге.

Стоит отметить, что основное предназначение библиотеки pandas – это подготовка данных. Однако библиотека pandas также предоставляет несколько функций для выполнения анализа данных. Эти возможности подразумевают вычисление описательных статистик и функций, необходимых для финансового анализа, например вычисления корреляции.

Поэтому сама по себе библиотека pandas не является инструментом для научных исследований. Это скорее инструмент обработки данных с некоторыми возможностями анализа. Библиотека pandas явно оставляет за скобками сложный статистический, финансовый анализ, предлагая его выполнить другим библиотекам Python, таким как SciPy, NumPy, scikit-learn, а для визуализации данных использует графические библиотеки, например **matplotlib** и **ggvis**.

Мы сосредоточимся на преимуществе библиотеки pandas над другими языками, такими как R, поскольку приложения на основе библиотеки pandas могут использовать обширную сеть надежных фреймворков Python, уже созданных и протестированных Python-сообществом.

Процесс анализа данных

Основная цель этой книги – научить вас использовать библиотеку pandas для обработки данных. Однако есть второстепенная и, возможно, не менее важная цель – показать, как библиотека pandas встроена в те процессы, которые специалист по работе с данными/аналитик данных выполняет в повседневной жизни.

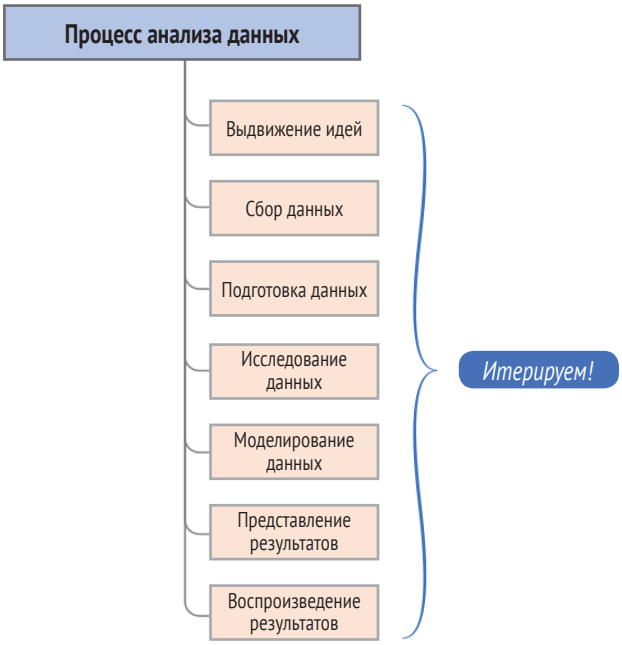
Описание шагов, связанных с процессом анализа данных, дается на веб-странице библиотеки pandas:

- обработка и очистка данных;
- анализ/моделирование данных;
- преобразование данных в удобную форму.

Этот небольшой список является хорошим исходным определением, но он не охватывает процесса анализа в целом и не может ответить на вопрос, почему в библиотеке pandas реализовано так много инструментов. В следующем разделе данный процесс будет рассмотрен подробнее.

Процесс

Предлагаемый процесс – это последовательность операций, которую называют процессом анализа данных, и он представлен на следующей диаграмме:



Эта схема задает структуру для определения логических шагов, которые будут предприняты в ходе работы с данными. Сейчас давайте кратко рассмотрим каждый этап этого процесса, а также некоторые задачи, которые вы как аналитик данных будете выполнять, используя библиотеку pandas.

Важно понимать, что это не простой линейный процесс. Лучше всего этот процесс осуществлять интерактивно и гибким/итеративным способом.

Выдвижение идей

Первый этап процесса анализа данных – определить, что вы хотите выяснить. Он называется **выдвижение идей** (**ideation**), мы формулируем идеи по поводу того, что мы хотим сделать и доказать. Идея в целом связана с гипотезой о наличии тех или иных структур в данных, которые можно использовать для принятия решений.

Эти решения часто принимаются в контексте бизнеса, а также в других дисциплинах, например в научно-исследовательской деятельности. Сейчас модно применять анализ данных для решения бизнес-задач, поскольку глубокое понимание данных может повысить прибыль компаний.

Однако какое решение мы обычно хотим принять? Ниже приводятся некоторые наиболее часто задаваемые вопросы:

- почему это происходит?
- можем ли мы предсказать будущее с использованием исторических данных?
- как я могу оптимизировать бизнес-операции в будущем?

Этот список ни в коем случае не является исчерпывающим, но он охватывает значительную часть причин, по которым проводится анализ данных. Чтобы получить ответы на эти вопросы, нужно осуществить сбор и анализ данных, связанных с решаемой проблемой. Необходимо определить, какие данные мы будем изучать, какова польза от исследования, как будут получены данные, каковы критерии успешности решения и как в конечном итоге информация будет представлена.

Сама по себе библиотека pandas не предлагает инструментов, позволяющих формулировать идеи. Но как только вы научитесь использовать библиотеку pandas, вы, естественно, поймете, как pandas может помочь вам в формулировании идей. Ведь к этому моменту вы вооружитесь мощным инструментом, который можно использовать для выдвижения множества сложных гипотез.

Сбор данных

Как только вы сформулировали идею, вы должны получить данные, чтобы попытаться проверить свою гипотезу. Эти данные могут быть собраны вашей организацией или получены от внешних поставщиков данных. Эти данные обычно поставляются в виде архивных данных или могут быть предоставлены в режиме реального времени (хотя pandas не так широко известен, как инструмент обработки данных в режиме реального времени).

Часто данные являются «сырыми», даже если они взяты из источников данных, созданных вами или вашей организацией. Понятие «сырые» означает, что данные могут быть неправильно сформированы, записаны в разных форматах или содержать ошибки. Они могут быть неполными, и может потребоваться аугментация данных[1].

В мире имеется множество бесплатных данных. Многие данные не являются бесплатными и фактически требуют значительных денежных сумм. Некоторые из них есть в свободном доступе и имеют публичные API-интерфейсы, а другие можно получить по подписке. Часто данные, которые вы покупаете, являются более чистыми, но это не всегда так.

В любом случае, библиотека pandas предлагает надежный и простой в использовании набор инструментов для сбора данных из разных источников, при этом данные могут иметь разный формат. Кроме того, библиотека pandas позволяет не только извлекать данные, но и выполнять первоначальную структуризацию данных с помощью объектов Series и DataFrame, не прибегая к написанию сложного

[1] Аугментация данных (data augmentation) – это создание дополнительных обучающих данных из имеющихся данных. – *Прим. перев.*

программного кода, который может потребоваться в других инструментах или языках программирования.

Подготовка данных

На этом этапе мы подготавливаем исходные данные для проведения дальнейшего исследования. Часто подготовка является весьма интересным этапом анализа данных. Как правило, все проблемы, связанные с данными, связаны с качеством этих данных. Вы, вероятно, потратите много времени на устранение проблем, связанных с качеством данных.

Зачем? Ну, есть ряд причин:

- данные просто некорректны;
- какая-то информация в наборе данных отсутствует;
- данные записаны в единицах измерения, не подходящих для вашего анализа;
- данные записаны в форматах, не удобных для вашего анализа;
- данные находятся на том уровне детализации, который не подходит для вашего анализа;
- не все интересующие вас поля можно получить из одного источника;
- представление данных отличается в зависимости от поставщика.

Процесс подготовки данных сосредоточен на решении вышеприведенных проблем. Библиотека pandas предлагает множество отличных возможностей для процесса подготовки данных, который часто называют **приведением данных в порядок** (**data tidying**). Ее арсенал включает в себя интеллектуальные средства обработки пропущенных данных, преобразование типов данных, преобразование форматов, изменение частоты измерений, объединение данных, расположенных в нескольких наборах данных, кодировку/преобразование символов и группировку данных среди прочих операций. Мы подробно рассмотрим все эти операции.

Исследование данных

Исследование данных подразумевает изучение ваших данных, чтобы попытаться открыть какие-то закономерности в данных. Исследование может включать в себя различные задачи, например:

- изучение того, как переменные связаны друг с другом;
- анализ распределения данных;
- обнаружение и исключение выбросов;
- создание быстрых визуализаций;
- быстрое создание новых представлений данных или моделей для последующего использования в ходе моделирования.

Исследование данных является одной из самых сильных сторон библиотеки pandas. Хотя большинство языков программирования также позволяет выполнить исследование данных, каждый из них требует подготовительных действий, не связанных непосредственно с исследованием данных.

Используя принцип **read-eval-print-loop** (**REPL**), реализованный в IPython и/или тетрадках Jupyter, библиотека pandas создает исследовательскую среду, в которой объем этих подготовительных действий сведен к минимуму. Выразительность синтаксиса библиотеки pandas позволяет кратко описать сложные операции с данными, а немедленно появляющийся результат выполненной операции

помогает быстро проверить правильность ваших действий, не прибегая к повторной компиляции и полному переписыванию своего программного кода.

Моделирование данных

На этапе моделирования вы формализуете свои открытия, найденные в ходе исследования данных. Для этого закономерностям, обнаруженным в данных и позволяющим извлечь из данных содержательный смысл, нужно дать ясную интерпретацию. Итогом становится **модель** (**model**), позволяющая выразить обнаруженные закономерности в математическом виде и содержащая программный код, позволяющий преобразовать данные в полезную информацию.

Процесс моделирования является итеративным, когда с помощью исследования данных вы выбираете переменные, необходимые для вашего анализа, подаете переменные на вход модели, строите модель и определяете, насколько хорошо модель подтверждает ваши исходные предположения. Он может включать формальное моделирование структуры данных, а также может сочетать методы из различных областей, например из статистики, машинного обучения и исследования операций.

Библиотека pandas предлагает мощные инструменты для моделирования данных. Именно на этом этапе вы формализуете модель данных, используя объекты DataFrame и стараясь сделать процесс построения модели максимально компактным. Кроме того, вы можете воспользоваться всеми возможностями языка Python, чтобы полностью автоматизировать процесс создания модели.

С аналитической точки зрения библиотека pandas предлагает в первую очередь интегрированную поддержку описательной статистики, которая понадобится вам при решении разных задач. Впрочем, библиотека pandas использует язык Python, и поэтому если вам нужны более продвинутые аналитические возможности, вы очень легко можете интегрировать pandas с другими библиотеками обширной научной среды Python.

Представление результатов

Предпоследний этап процесса анализа данных – представление результатов, как правило, в виде отчета или презентации. Вам нужно дать убедительное и подробное объяснение своего решения. Часто это можно сделать с помощью различных инструментов визуализации в Python и затем вручную создать презентацию.

Тетрадки (ноутбуки) Jupiter – это мощный инструмент для создания презентаций по итогам вашего анализа. Эти тетрадки представляют собой инструмент как для выполнения кода, так и для подробного аннотирования выполняемых действий. Их можно использовать для создания высокоэффективных презентаций, включающих фрагменты программного кода, стилизованный текст и графику.

Мы кратко рассмотрим тетрадки Jupyter в главе 2 «Запуск библиотеки pandas».

Воспроизведение результатов анализа

Важный этап процесса анализа данных – обмен и воспроизведение результатов. Часто говорят, что если другие исследователи не смогут воспроизвести ваш эксперимент и полученные результаты, то вы ничего не доказали.

К счастью для вас, вы легко сможете сделать свой анализ воспроизводимым, воспользовавшись библиотекой pandas и Python. Это можно сделать, поделившись с другими исследователями программным кодом Python, который лежит в основе программного кода библиотеки pandas, а также данными.

Тетрадки Jupyter также являются удобным инструментом для хранения программного кода и проектов, которыми можно легко поделиться со всеми, у кого установлен Jupyter Notebook. В интернете существует много бесплатных и безопасных сайтов обмена, которые позволяют вам создавать или размещать свои тетрадки Jupyter для совместного использования.

По поводу итеративности и гибкости

Очень важно понимать, что обработка данных, анализ и наука – это итеративный процесс. Несмотря на то что ранее мы рассмотрели этапы, выстроив прямую последовательность, в конечном итоге вы будете двигаться как в прямом, так и в обратном порядке. Например, на этапе исследования вы можете выявить аномалии в данных, которые связаны с проблемами чистоты данных, они решаются на этапе подготовки, и поэтому необходимо вернуться, чтобы исправить эти проблемы.

Это обычная ситуация, возникающая в процессе анализа данных. Вы словно совершаете путешествие, пытаетесь решить свою первоначальную задачу, все время получая новое, дополнительное представление о данных, с которыми работаете. Эта информация, возможно, побудит вас задать новые вопросы, более точные вопросы или осознать, что ваши первоначальные вопросы не являлись теми вопросами, на которые вы хотели бы найти ответы. Процесс анализа данных – это и в самом деле путешествие, а не конкретный пункт назначения.

Взаимосвязь между книгой и процессом анализа данных

Ниже приведено краткое описание этапов анализа данных, о которых вы узнаете в этой книге. Не волнуйтесь, что последовательность этапов не совпадает с последовательностью изложения материала в книге. Книга в логическом порядке знакомит вас с библиотекой pandas, и вы можете вернуться к любой главе, которая соответствует интересующему вас этапу анализа данных.

Этап процесса анализа данных	Глава книги
Выдвижение идей	Выдвижение идей – это самая творческая часть науки о данных. У вас должна быть идея. Она должна подверждаться фактами (здесь вам понадобятся данные), и должна быть возможность наблюдать эти факты после проведения анализа (на основе фактов из прошлого строим модель, прогнозируя их в будущем)
Сбор данных	Сбор данных главным образом освещается в главе «Загрузка данных»
Подготовка данных	Подготовка данных преимущественно освещается в главе 10 «Приведение данных в порядок», но это довольно общая тема, которая встречается в большинстве глав данной книги
Исследование данных	Тема исследования данных охватывает материал, начинающий с главы 3 «Представление одномерных данных с помощью объекта Series» и заканчивающийся главой 14 «Визуализация». Таким образом, большая часть книги посвящена исследованию данных. Но наиболее подробное освещение данного этапа дается в главе 14 «Визуализация»

Этап процесса анализа данных	Глава книги
Моделирование данных	Моделирование данных подробно освещается в главе 3 «Представление одномерных данных с помощью объекта Series» и главе 4 «Представление табличных и многомерных данных с помощью объекта DataFrame», а также в главе 11 «Объединение, связывание и изменение формы данных» и главе 13 «Анализ временных рядов»
Представление результатов	Представление результатов – это главная тема главы 14 «Визуализация»
Воспроизведение результатов	Тема воспроизведения результатов проходит по всей книге, поскольку примеры представлены в виде тетрадок Jupyter. Работая в тетрадках, вы по умолчанию используете инструмент для воспроизведения результатов, и у вас есть возможность поделиться этими тетрадками различными способами

Понятия «данные» и «анализ» в контексте нашего знакомства с библиотекой pandas

По мере изучения библиотеки pandas и анализа данных вы будете сталкиваться с разными понятиями из области данных, моделирования и анализа. Давайте рассмотрим некоторые из этих понятий и выясним, как они связаны с библиотекой pandas.

Типы данных

При работе с сырыми данными вы столкнетесь с несколькими типами данных, которые нужно будет представить с помощью объектов pandas. Они важны для понимания, поскольку инструменты, необходимые для работы с каждым типом, отличаются друг от друга.

Библиотека pandas, по сути, используется для обработки структурированных данных. Однако, помимо этого, она предлагает несколько инструментов, позволяющих преобразовать неструктурированные данные в такое представление, которое мы уже сможем обработать.

Структурированные данные

Структурированные данные – это тип данных, при котором последние упорядочены в вертикальные столбцы (поля) и горизонтальные строки (записи или наблюдения). Примером таких данных являются данные в реляционных базах и электронных таблицах. Структурированные данные зависят от модели данных, которая представляет собой определенную структуру, содержательного смысла данных и часто от способа обработки данных. Сюда входят идентификация типа данных (целое число, число с плавающей точкой, строка и т. д.) и ограничения, накладываемые на данные, например количество символов, максимальное и минимальное значения, или ограничение на определенный набор значений.

Структурированный данные – это тип данных, который предназначен для использования в библиотеке pandas. Как мы увидим сначала на примере объекта `Series`, а затем на примере объекта `DataFrame`, pandas помещает структурированные данные в один или несколько столбцов данных (при этом каждый столбец имеет один конкретный тип данных) и одну или несколько строк данных.

Неструктурированные данные

Неструктурированные данные – это данные, которые не имеют определенной структуры, не предполагают наличия заранее определенных столбцов определенного типа. Примером неструктурированных данных могут быть такие виды информации, как фотографии и графические изображения, видеоролики, потоковые данные датчиков, веб-страницы, PDF-файлы, презентации PowerPoint, электронные письма, записи в блогах, страницы Википедии и документы Word.

Хотя библиотека pandas не может обработать неструктурированные данные напрямую, она предлагает ряд инструментов для сбора структурированных данных из неструктурированных источников. В качестве конкретного примера мы рассмотрим инструменты pandas, позволяющие извлекать веб-страницы, определенные фрагменты контента и записывать в объект `DataFrame`.

Полуструктурированные данные

Полуструктурированные данные занимают промежуточное положение между неструктурированными и структурированными данными. Их можно рассматривать как тип структурированных данных, у которых нет строгой структуры моделей данных. Формат JSON – это пример полуструктурированных данных. Хотя хороший JSON-файл будет иметь определенный формат, здесь нет определенной схемы данных, которая всегда строго соблюдается. В большинстве случаев данные будут записаны в виде определенной схемы, которую легко преобразовать в структурированный тип данных, например в объект `DataFrame`, но для этого, возможно, потребуется задать определенный тип данных.

Переменные

Моделируя данные в библиотеке pandas, мы будем исследовать одну или несколько переменных и искать статистический смысл, анализируя значения этой переменной или значения нескольких переменных. При этом термин «переменная» не тождествен понятию «переменная» в программировании, здесь речь идет о статистических переменных.

Переменной является любая характеристика, число или количество, которое можно измерить или сосчитать. Название «переменная» обусловлено тем, что значение характеристики может меняться от наблюдения к наблюдению, а также может меняться со временем. Значение цены акции, возраст, пол, доходы и расходы в бизнесе, страна рождения, капитальные затраты, школьные оценки, цвет глаз и тип транспортного средства являются примерами переменных.

Существует несколько общих типов статистических переменных, с которыми мы столкнемся при работе с библиотекой pandas:

- категориальные переменные;
- непрерывные переменные.

Категориальные переменные

Категориальная переменная (**categorical variable**) – это переменная, которая может принимать одно значение из ограниченного и обычно фиксированного набора возможных значений. Каждое из возможных значений часто называется **уровнем** (**level**). Примерами категориальных переменных являются пол, штат, социальный класс, группа крови, гражданство, время наблюдения или рейтинг, например шкала Лайкерта. Все категориальные переменные являются качествен-

ными характеристиками, которые могут описать продукт, но при этом не измеряются в непрерывной шкале. Например, возьмем переменную *Штат*. Допустим, он имеет уровни Аризона, Северная Каролина и Висконсин. Человек либо проживает в штате Аризона, либо в штате Северная Каролина, либо в штате Висконсин. Не существует золотой середины между штатами, нельзя вычислить среднее значение штата, и нет естественного способа упорядочить эти категории, нельзя сказать, что штат Аризона хуже/лучше штата Северная Каролина, а штат Северная Каролина хуже/лучше штата Висконсин. Категориальные переменные в библиотеке pandas представлены объектами `Categorical`, специальным типом данных, который соответствует категориальным переменным в статистике.

Непрерывные переменные

Непрерывная переменная (**continuous variable**) – это переменная, которая может принимать бесконечное (неисчислимое) количество значений. Все непрерывные переменные являются характеристиками, которые количественно описывают продукт и измеряются в непрерывной шкале. Примерами непрерывных переменных являются возраст, высота, время и температура. Мы можем вычислить средний возраст, среднюю высоту, среднее время и среднюю температуру. Мы можем упорядочить значения. 20-летний младше 30-летнего, а 30-летний младше 40-летнего. Непрерывные переменные в библиотеке pandas представлены либо типом `float`, либо типом `integer` (нативными питоновскими типами данных).

Временные ряды

В библиотеке pandas временные ряды – это объект первого класса. Время добавляет важное дополнительное измерение. Часто переменные не зависят от времени их регистрации, то есть момент времени, в который их фиксировали, не имеет значения. Но во многих случаях это не так. Временной ряд представляет собой переменную, у которой значения зарегистрированы в определенные временные интервалы, таким образом, наблюдения изначально упорядочены по времени.

Стохастическая модель для временного ряда обычно отражает тот факт, что наблюдения, близкие друг к другу во времени, будут более тесно связаны, чем наблюдения, которые далеко отстоят друг от друга по времени. Модели временных рядов часто используют естественное одностороннее упорядочение времени, поэтому значения для заданного периода будут выражаться как полученные из прошлого, а не из будущего.

Чаще всего в библиотеке pandas работают с финансовыми данными, где переменная представляет собой стоимость акции, изменяющуюся через равные интервалы времени в течение дня. Нам часто нужно определить скорость изменения цены в определенные интервалы времени. Кроме того, мы можем скорректировать цены нескольких акций по определенным интервалам времени.

Это настолько важная и мощная возможность библиотеки pandas, что мы посвятим ей целую главу.

Общие понятия анализа и статистики

В этой книге мы только коснемся общих понятий статистики и технической стороны анализа данных. Однако стоит раскрыть несколько понятий, некоторые из которых реализованы непосредственно в библиотеке pandas. Остальные понятия

связаны с другими библиотеками, например с библиотекой SciPy, однако вы также можете встретить их во время работы с библиотекой pandas, поэтому мы тоже о них расскажем.

Количественный и качественный анализ

Качественный анализ – это научное исследование данных, которые можно наблюдать, но нельзя количественно измерить. Основное внимание уделяется качественным характеристикам. Примерами качественных данных могут быть:

- мягкость кожи;
- изящность бега.

Количественный анализ – это исследование данных, которые можно измерить количественно. Примерами количественных данных могут быть:

- количество;
- цена;
- высота.

Библиотека pandas главным образом работает с количественными данными, предлагая разнообразные инструменты для их обработки. Библиотека pandas не предназначена для проведения качественного анализа, но позволяет вам представить качественную информацию в различных видах.

Одномерный и многомерный анализ

В каком-то смысле статистика представляет собой практику изучения переменных и, в частности, наблюдение за этими переменными. Статистика преимущественно опирается на анализ одной переменной, которая называется одномерным анализом. **Одномерный анализ** (**univariate analysis**) – это простейшая форма анализа данных. Он не имеет отношения к анализу причин или взаимосвязей и обычно используется для описания или подытоживания данных, а также поиска закономерностей в данных.

Многомерный анализ (**multivariate analysis**) – это метод моделирования, в котором участвуют две или более переменных, влияющих на результат эксперимента. Многомерный анализ часто связан с такими понятиями, как корреляция и регрессия, которые помогают нам понять взаимосвязь между несколькими переменными, а также влияние этой взаимосвязи на результат.

Библиотека pandas в основном предлагает инструменты для проведения одномерного анализа. Главным образом речь идет о вычислении описательных статистик, хотя можно вычислить такой показатель, как корреляция (поскольку ее часто используют в финансах и других областях).

Остальные более сложные статистические процедуры можно выполнить с помощью библиотеки StatsModels. Опять же это не недостаток библиотеки pandas, а конкретное решение, направленное на то, чтобы более сложные операции выполнять с помощью специализированных библиотек Python.

Описательные статистики

Описательные статистики – это показатели, которые подытоживают данные, обычно в тех случаях, где набор данных представляет собой генеральную совокупность или выборку из одной переменной (одномерные данные). Эти показатели описывают набор данных и являются мерами центральной тенденции, а также мерами изменчивости и дисперсии.

Например, следующие показатели являются описательными статистиками:

- распределение (например, нормальное, пуассоновское);
- центральная тенденция (например, среднее, медиана и мода);
- дисперсия (например, дисперсия, стандартное отклонение).

Позже мы увидим, что объекты `Series` и `DataFrame` поддерживают вычисление большей части описательных статистик.

Индуктивные статистики

Индуктивные статистики отличаются от описательных тем, что с их помощью мы пытаемся сделать выводы о данных, а не просто подытожить данные. Примерами индуктивных статистик являются:

- t-тест;
- хи-квадрат;
- ANOVA;
- бутстреп.

Эти индуктивные методы были перенесены из библиотеки pandas в другие инструменты типа SciPy и StatsModels.

Стохастические модели

Стохастические модели представляют собой вид статистического моделирования, который включает в себя одну или несколько случайных величин, а также подразумевает использование временных рядов. Цель стохастической модели – оценить вероятность того, что результат будет находиться в пределах определенного интервала, и спрогнозировать условия для разных ситуаций.

Примером стохастического моделирования являются симуляции Монте-Карло. Симуляции Монте-Карло применяются для расчета стоимости финансовых портфелей, которые подвержены влиянию различных риск-факторов и имеют нетривиальные распределения доходности.

Библиотека pandas предлагает фундаментальную структуру данных DataFrame, ее часто используют при работе с временными рядами, чтобы на их основе потом создать и запустить стохастическую модель. Хотя можно создать свои собственные стохастические модели и провести анализ с помощью pandas и Python, во многих случаях более удобным инструментом для стохастического моделирования будут специализированные библиотеки типа PyMC.

Вероятность и байесовская статистика

Байесовская статистика – это подход к статистическому оцениванию, основанный на теореме Байеса, математическом уравнении, использующем простые аксиомы теории вероятностей. Она позволяет аналитику вычислить любую условную вероятность интересующего события. Условная вероятность – это просто вероятность события A, при условии что наступило событие B.

Поэтому если использовать термины теории вероятностей, события уже произошли и были зафиксированы (поскольку мы знаем вероятность). Используя теорему Байеса, мы можем вычислить вероятность различных интересующих нас событий, учитывая уже имеющиеся данные.

Байесовское моделирование выходит за рамки этой книги, но опять же соответствующие модели данных прекрасно обрабатываются с помощью библиотеки pandas, и затем их можно анализировать с помощью таких библиотек, как PyMC.