

Содержание

Предисловие	13
Глава 1. Библиотека pandas и анализ данных	17
Знакомство с библиотекой pandas	17
Обработка данных, анализ, наука и библиотека pandas	19
Обработка данных	20
Анализ данных.....	20
Наука о данных	21
Предназначение библиотеки pandas.....	21
Процесс анализа данных.....	21
Процесс.....	22
Взаимосвязь между книгой и процессом анализа данных.....	26
Понятия «данные» и «анализ» в контексте нашего знакомства с библиотекой pandas	27
Типы данных.....	27
Переменные	28
Категориальные переменные	28
Временные ряды.....	29
Общие понятия анализа и статистики.....	29
Другие библиотеки Python, работающие вместе с библиотекой pandas	32
Численные и научные вычисления – NumPy и SciPy	32
Статистический анализ – StatsModels.....	33
Машинное обучение – scikit-learn.....	33
РуМС – стохастическое байесовское моделирование	33
Визуализация данных – matplotlib и seaborn	34
Выводы	34
Глава 2. Запуск библиотеки pandas	36
Установка Anaconda	36
IPython и Jupyter Notebook.....	39
IPython.....	39
Jupyter Notebook.....	40
Знакомство со структурами данных библиотеки pandas – Series и DataFrame	43
Импорт pandas	44
Объект Series	45
Объект DataFrame	49
Загрузка данных из CSV-файла в объект DataFrame	54
Визуализация	57
Выводы	57

Глава 3. Представление одномерных данных с помощью объекта Series	58
Настройка библиотеки pandas	59
Создание объекта Series	59
Создание объекта Series с помощью питоновских списков и словарей	59
Создание объекта Series с помощью функций NumPy	61
Создание объекта Series с помощью скалярного значения	62
Свойства <code>.index</code> и <code>.values</code>	63
Размер и форма объекта Series	63
Установка индекса во время создания объекта Series	64
Использование методов <code>.head()</code> , <code>.tail()</code> и <code>.take()</code> для вывода значений	65
Получение значений в объекте Series по метке или позиции	66
Поиск по метке с помощью оператора <code>[]</code> и свойства <code>.ix[]</code>	66
Явный поиск по позиции с помощью свойства <code>.iloc[]</code>	68
Явный поиск по меткам с помощью свойства <code>.loc[]</code>	68
Создание срезов объекта Series	69
Выравнивание данных по меткам индекса	74
Выполнение логического отбора	77
Переиндексация объекта Series	79
Модификация объекта Series на месте	83
Выводы	85
Глава 4. Представление табличных и многомерных данных с помощью объекта DataFrame	86
Настройка библиотеки pandas	87
Создание объектов DataFrame	87
Создание объекта DataFrame на основе результатов функций NumPy	87
Создание объекта DataFrame с помощью питоновского словаря и объектов Series	89
Создание объекта DataFrame на основе CSV-файла	91
Доступ к данным внутри объекта DataFrame	92
Отбор столбцов в объекте DataFrame	93
Отбор строк в объекте DataFrame	94
Поиск скалярного значения по метке и позиции с помощью <code>.at[]</code> и <code>.iat[]</code>	95
Создание среза датафрейма с помощью оператора <code>[]</code>	96
Логический отбор строк	96
Одновременный отбор строк и столбцов	98
Выводы	99
Глава 5. Выполнение операций над объектом DataFrame и его содержимым	100
Настройка библиотеки pandas	100
Переименование столбцов	101
Добавление новых столбцов с помощью оператора <code>[]</code> и метода <code>.insert()</code>	102
Добавление столбцов за счет расширения датафрейма	103
Добавление столбцов с помощью конкатенации	104

Переупорядочивание столбцов	105
Замена содержимого столбца	106
Удаление столбцов	106
Присоединение новых строк	108
Конкатенация строк	110
Добавление и замена строк за счет расширения датафрейма	111
Удаление строк с помощью метода .drop()	112
Удаление строк с помощью логического отбора	112
Удаление строк с помощью среза	113
Выводы	114
Глава 6. Индексация данных	115
Настройка библиотеки pandas	115
Важность применения индексов	116
Типы индексов библиотеки pandas	118
Основной тип Index	118
Индексы Int64Index и RangeIndex, в качестве меток используются целые числа	118
Индекс Float64Index, в качестве меток используются числа с плавающей точкой	120
Представление дискретных интервалов с использованием IntervalIndex	120
Категории в качестве индекса – CategoricalIndex	121
Индексация по датам и времени с помощью DatetimeIndex	122
Индексация периодов времени с помощью PeriodIndex	122
Работа с индексами	123
Создание и использование индекса в объекте Series или объекте DataFrame	123
Отбор значений с помощью индекса	124
Преобразование данных в индекс и получение данных из индекса	126
Переиндексация объекта библиотеки pandas	127
Иерархическая индексация	128
Выводы	131
Глава 7. Категориальные данные	132
Настройка библиотеки pandas	132
Создание категориальных переменных	133
Переименование категорий	138
Добавление категорий	139
Удаление категорий	139
Удаление неиспользуемых категорий	140
Установка категорий	140
Вычисление описательных статистик для категориальной переменной	141
Обработка школьных оценок	141
Выводы	144
Глава 8. Численные и статистические методы	145
Настройка библиотеки pandas	146
Применение численных методов к объектам библиотеки pandas	146

Выполнение арифметических операций над объектами DataFrame или Series.....	147
Вычисление количества значений.....	150
Определение уникальных значений (и их встречаемости).....	150
Вычисление минимума и максимума.....	151
Вычисление n наименьших значений и n наибольших значений.....	151
Вычисление накопленных значений.....	152
Выполнение статистических операций с объектами библиотеки pandas.....	153
Получение итоговых описательных статистик.....	153
Измерение центральной тенденции: среднее, медиана и мода.....	154
Вычисление дисперсии и стандартного отклонения.....	156
Вычисление ковариации и корреляции.....	157
Дискретизация и квантилизация данных.....	159
Вычисление ранга значений.....	163
Вычисление процентного изменения для каждого наблюдения серии.....	164
Выполнение операций со скользящим окном.....	164
Создание случайной выборки данных.....	167
Выводы.....	168
Глава 9. Загрузка данных.....	169
Настройка библиотеки pandas.....	170
Работа с CSV-файлами и текстовыми/табличными данными.....	170
Исследование CSV-файла.....	170
Чтение CSV-файла в датафрейм.....	171
Указание индекса столбца при чтении CSV-файла.....	171
Вывод и спецификация типа данных.....	172
Указание имен столбцов.....	173
Указание конкретных столбцов для загрузки.....	173
Сохранение датафрейма в CSV-файл.....	174
Работа с данными, в которых используются разделители полей.....	174
Обработка загрязненных данных, в которых используются разделители полей.....	175
Чтение и запись данных в формате Excel.....	177
Чтение и запись JSON-файлов.....	180
Чтение HTML-файлов из Интернета.....	181
Чтение и запись HDF5-файлов.....	183
Загрузка CSV-файлов из Интернета.....	185
Чтение из базы данных SQL и запись в базу данных SQL.....	186
Загрузка данных с удаленных сервисов.....	188
Загрузка данных о котировках акций с веб-сервисов Yahoo! и Google Finance.....	189
Загрузка данных об опционах с веб-сервиса Google Finance.....	189
Загрузка Базы данных по экономической статистике Федерального резервного банка Сент-Луиса.....	191
Загрузка данных Кеннета Френча.....	193
Загрузка данных Всемирного банка.....	193
Выводы.....	197

Глава 10. Приведение данных в порядок	198
Настройка библиотеки pandas	198
Что такое приведение данных в порядок?.....	199
Как работать с пропущенными данными	199
Поиск значений NaN в объектах библиотеки pandas.....	201
Удаление пропущенных данных	203
Обработка значений NaN в ходе арифметических операций.....	206
Заполнение пропущенных данных	207
Прямой и обратный порядки заполнения пропущенных значений.....	208
Заполнение с помощью меток индекса	209
Выполнение интерполяции пропущенных значений	210
Обработка дублирующихся данных	212
Преобразование данных	214
Сопоставление значений другим значениям.....	215
Замена значений	216
Применение функций для преобразования данных	219
Выводы	222
Глава 11. Объединение, связывание и изменение формы данных	223
Настройка библиотеки pandas	223
Конкатенация данных, расположенных в нескольких объектах.....	224
Понимание семантики конкатенации, принятой по умолчанию.....	224
Переключение осей выравнивания.....	227
Определение типа соединения.....	229
Присоединение вместо конкатенации	230
Игнорирование меток индекса	230
Слияние и соединение данных.....	231
Слияние данных, расположенных в нескольких объектах	231
Настройка семантики соединения при выполнении слияния.....	234
Поворот данных для преобразования значений в индексы и наоборот	236
Состыковка и расстыковка данных	238
Состыковка с помощью неиерархических индексов	238
Расстыковка с помощью иерархических индексов.....	239
Расплавление данных для преобразования «широкого» формата в «длинный» и наоборот	243
Преимущества использования состыкованных данных.....	244
Выводы	245
Глава 12. Агрегирование данных	246
Настройка библиотеки pandas	246
Обзор схемы «разделение – применение – объединение».....	247
Данные для примеров	248
Разделение данных.....	248
Группировка по значениям отдельного столбца.....	248
Просмотр результатов группировки	249
Группировка по нескольким столбцам	252
Группировка по уровням индекса	253

Применение агрегирующих функций, преобразований и фильтров	255
Применение агрегирующих функций к группам.....	255
Преобразование групп данных.....	257
Исключение групп из процедуры агрегирования.....	262
Выводы	263
Глава 13. Анализ временных рядов	264
Настройка библиотеки pandas	264
Представление дат, времени и интервалов	265
Объекты datetime, day и time	265
Создание временной метки с помощью объекта Timestamp	267
Использование объекта Timedelta для представления временного интервала	267
Введение во временные ряды.....	268
Индексация с помощью объекта DatetimeIndex.....	268
Создание временного ряда с определенной частотой.....	273
Вычисление новых дат с помощью смещений	275
Представление временных интервалов с помощью смещений дат	275
Привязанные смещения	278
Представление промежутков времени с помощью объектов Period	279
Создание временного интервала с помощью объекта Period	279
Индексация с помощью объекта PeriodIndex	281
Обработка праздников с помощью календарей.....	283
Нормализация временных меток с помощью часовых поясов.....	284
Операции с временными рядами.....	288
Опережение и запаздывание.....	288
Преобразование частоты временного ряда.....	291
Увеличение или уменьшение шага дискретизации временного ряда	293
Применение к временному ряду операций на основе скользящего окна	298
Выводы	301
Глава 14. Визуализация	302
Настройка библиотеки pandas	303
Основные инструменты визуализации	303
Создание графиков временных рядов	304
Настройка внешнего вида графика временного ряда	306
Виды графиков, часто использующиеся в статистическом анализе данных	317
Демонстрация относительных различий с помощью столбиковых диаграмм.....	317
Визуализация распределений данных с помощью гистограмм	319
Визуализация распределений категориальных данных с помощью ящичных диаграмм с усами.....	321
Отображение накопленных итогов с помощью площадных диаграмм	322
Визуализация взаимосвязи между двумя переменными с помощью диаграммы рассеяния	323
Визуализация оценок распределения с помощью графика ядерной оценки плотности.....	324

Визуализация корреляций между несколькими переменными с помощью матрицы диаграмм рассеяния	325
Отображение взаимосвязей между несколькими переменными с помощью тепловых карт	326
Размещение нескольких графиков на одном рисунке вручную	327
Выводы	330
Глава 15. Анализ исторических котировок акций	331
Настройка библиотеки pandas	331
Загрузка и изменение формы данных о котировках акций, предоставляемых веб-сервисом Yahoo!Finance	332
Визуализация временных рядов котировок акций.....	335
Визуализация временных рядов объемов торгов	336
Вычисление простого ежедневного процентного изменения цены закрытия.....	338
Вычисление простой ежедневной накопленной доходности акции	339
Изменение шага дискретизации временного ряда для вычисления ежемесячной доходности вместо ежедневной	341
Анализ распределения доходности.....	343
Выполнение вычислений на основе скользящего среднего.....	344
Сравнение средних ежедневных доходностей акций.....	345
Корреляция между котировками на основе ежедневного процентного изменения цены закрытия	348
Вычисление волатильности акций.....	349
Визуализация взаимосвязи между риском и ожидаемой доходностью.....	351
Выводы	352
Приложение 1. Советы по оптимизации вычислений в библиотеке pandas	353
Базовое итерирование	354
Итерирование с помощью метода .iterrows()	355
Лучший способ итерирования с помощью метода .apply()	355
Векторизация с помощью объектов Series	356
Векторизация с помощью массивов NumPy.....	356
Выводы	357
Приложение 2. Улучшение производительности pandas (из официального пособия по библиотеке pandas)	358
Написание расширений на языке C для pandas	358
«Чистый» Python	358
Обычный Cython.....	360
Использование библиотеки Numba.....	360
Jit.....	361
Vectorize.....	361
Вычисление выражений с помощью функции eval()	363
Поддерживаемый синтаксис	363

Примеры использования функции eval().....	364
Метод DataFrame.eval().....	364
Приложение 3. Используем pandas для больших данных	368
Работаем с данными бейсбольных игр	368
Внутреннее представление датафрейма	370
Подтипы	371
Оптимизация числовых столбцов с помощью понижающего преобразования	372
Сравнение способов хранения числовых и строковых значений.....	374
Оптимизация типов object с помощью типа category	376
Задаем типы во время считывания данных	380
Задаем типы во время считывания данных	382
Приложение 4. Пример предварительной подготовки данных в pandas	383
Считывание CSV-файла в объект DataFrame	384
Преобразование типов переменных	385
Переименование категорий переменных	387
Обработка редких категорий.....	388
Разбиение набора данных на обучающую и контрольную.....	392
Импутация пропусков.....	394
Конструирование новых признаков.....	399
Создание переменной, у которой значения основаны на значениях исходной переменной.....	400
Создание переменной, у которой каждое значение – частота наблюдений в категории переменных	402
Создание переменной, у которой каждое значение – среднее значение количественной переменной, взятое по уровню категориальной переменной	403
Создание категориальной переменной в результате конъюнкции двух переменных	403
Создание количественной переменной, у которой каждое значение – усредненное значение нескольких количественных переменных.....	404
Категоризация (биннинг) количественной переменной	404
Создание бинарной переменной на основе значений количественной переменной.....	406
Дамми-кодирование и подготовка массивов для обучения и проверки.....	408
Построение случайного леса и логистической регрессии	411
Отдельная предварительная подготовка данных для логистической регрессии.....	414

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 Jupyter.

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 и 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). Мы рассмотрим создание нескольких распространенных типов графиков, которые используются для представления различных типов данных.

Глава 15 «Анализ исторических котировок акций» расскажет, как применять библиотеку `pandas` для решения основных финансовых задач. Она будет посвящена данным, загруженным с веб-сервиса `Yahoo!Finance`, и в ней будет продемонстрирован расчет различных финансовых показателей, таких как доходность, скользящие средние, волатильность и др. Читатель также научится визуализировать эти финансовые показатели.

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 Python 2.7` и `pandas 0.15.1`. Если вы не установили их, в главе 2 «Запуск библиотеки `pandas`» дается инструкция относительно установки `pandas` в системах `Windows`, `OSX` и `Ubuntu`. Для тех, кто не хочет устанавливать какое-либо программное обе-

спечение, также дается инструкция по использованию онлайн-сервиса Warkari.io, предназначенного для анализа данных с помощью языка Python.

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 этой 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 следующим 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. Слова, которые вы видите на экране, например в меню или диалоговых окнах, отображаются в тексте следующим образом: «Ярлыки в этой книге используют схему Mac OS X 10.5+».



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 Ракст 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 веб-сайта, 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 поддержка 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 содержательно-го 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 встроеной 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 были 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, 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 данных. 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 этих 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 этот 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 при-

менять 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 данных¹.

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

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

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, обусловлены 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 выполнять 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 исследование 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 Jupyter – 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 в 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 подтверждаться 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 этой книги
Исследование данных	Тема исследования данных охватывает материал, начинающийся с главы 3 «Представление одномерных данных с помощью объекта Series» и заканчивающийся главой 15 «Анализ исторических котировок акций». Таким образом, большая часть книги посвящена исследованию данных. Но наиболее подробное освещение данного этапа дается в главе 14 «Визуализация» и главе 15 «Анализ исторических котировок акций». В обеих главах мы ознакомимся с результатами нашего анализа
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». В главе 15 «Анализ исторических котировок акций» разбирается моделирование данных применительно к финансам

Этап процесса анализа данных	Глава книги
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 «ДААННЫЕ» 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.

Структурированные 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, по-

токовые 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 конкретной 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 качественными 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 про-

живает 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 (неисчислимо) 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 анализы

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 анализы

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, которые влияют 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 `RyMC`.

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.

Корреляция

Корреляция является одной из наиболее распространенных статистик, и ее можно вычислить с помощью соответствующего метода объекта DataFrame. Корреляция – это число, которое описывает степень взаимосвязи между двумя переменными.

Распространенным примером использования корреляции является определение того, насколько сильно цены двух акций меняются вместе с течением времени. Если изменения сильно связаны, эти две акции имеют высокую корреляцию, если нет – между ними нет корреляции. Это ценная информация, которую можно использовать в ряде инвестиционных стратегий.

Кроме того, корреляция двух акций может зависеть от временного периода, который охватывает набор данных, а также от интервала внутри этого временного периода. К счастью, библиотека pandas обладает мощными возможностями, позволяющими учесть это и повторно вычислить корреляцию. Мы рассмотрим корреляции в нескольких главах.

Регрессия

Регрессия – это статистический метод, который оценивает силу взаимосвязи между зависимой переменной и рядом других переменных. Ее можно использовать для анализа взаимосвязей между переменными. Пример из области финансов – анализ взаимосвязи между ценами на товары и стоимостью акций предприятий, занимающихся производством этих товаров.



Первоначально библиотека pandas включала возможность построения регрессии, но затем она была перенесена в библиотеку StatsModels. Данный факт хорошо иллюстрирует процесс разработки библиотеки pandas. Часто в библиотеке pandas уже есть встроенные методы, но когда они достигают стадии «зрелости», принимается решение, что эффективнее будет перенести их в другие специализированные библиотеки Python. Это и хорошо, и плохо. Поначалу здорово иметь такую возможность непосредственно в pandas, но по мере обновления версий подобное нагромождение методов может сильно усложнить программный код!

ДРУГИЕ БИБЛИОТЕКИ PYTHON, РАБОТАЮЩИЕ ВМЕСТЕ С БИБЛИОТЕКОЙ PANDAS

Библиотека pandas представляет собой небольшую, но важную часть экосистемы анализа данных и науки о данных в Python. Для справки вот еще несколько важных библиотек Python, которые стоит отметить. Список не является исчерпывающим, но включает те библиотеки, с которыми вы, вероятно, столкнетесь.

Численные и научные вычисления – NumPy и SciPy

NumPy (<http://www.numpy.org/>) – базовый набор инструментов для осуществления научных вычислений с помощью Python, включенный в большинство современных дистрибутивов Python. Это действительно основной инструментарий, на базе

которого была создана библиотека pandas, и, работая с pandas, вы почти наверняка будете часто использовать NumPy. Среди прочего NumPy обеспечивает поддержку многомерных массивов, базовые операции над ними и полезные функции линейной алгебры.

Использование функций NumPy идет рука об руку с использованием библиотеки pandas, в частности когда мы работаем с объектом Series. Большинство наших примеров будет ссылаться на NumPy, но функциональность объекта Series – это настолько сложная надстройка над массивом NumPy, в детали которой мы вникать не будем, за исключением лишь некоторых ситуаций.

SciPy (<https://www.scipy.org/>) предлагает набор численных алгоритмов и инструментов для конкретных научных областей, включая обработку сигналов, оптимизацию, статистику и многое другое.

Статистический анализ – StatsModels

StatsModels (<http://statsmodels.sourceforge.net/>) – это модуль Python, который позволяет пользователям исследовать данные, строить статистические модели и выполнять статистические тесты. Широкий выбор описательных статистик, статистических тестов, графических функций доступен для разных типов данных и моделей. Исследователи разных областей могут обнаружить, что StatsModels полностью удовлетворяет их потребности в статистических вычислениях и анализе данных в среде Python.

Функционал StatsModels включает в себя:

- линейные регрессионные модели;
- обобщенные линейные модели;
- модели дискретного выбора;
- робастные линейные модели;
- различные модели и функции для анализа временных рядов;
- непараметрические оценки;
- коллекции наборов данных в качестве примеров;
- широкий спектр статистических тестов;
- инструменты ввода-вывода для создания таблиц в различных форматах (текст, LaTeX, HTML) и для чтения файлов Stata в массивы NumPy и дата-фреймы pandas;
- графические функции;
- различные модульные тесты для проверки правильности результатов.

Машинное обучение – scikit-learn

scikit-learn (<http://scikit-learn.org/>) – это библиотека машинного обучения, созданная на основе NumPy, SciPy и matplotlib. Она предлагает простые и эффективные инструменты для общих задач анализа данных, таких как классификация, регрессия, кластеризация, снижение размерности, отбор модели и предварительная обработка.

PyMC – стохастическое байесовское моделирование

PyMC (<https://github.com/pymc-devs/pymc>) – это модуль Python, в котором реализованы байесовские статистические модели и алгоритмы подгонки, включая метод

марковских цепей Монте-Карло. Гибкость и расширяемость этого модуля позволяют применить его к широкому кругу задач. Наряду с базовым функционалом – алгоритмами генерирования выборки – PyMC включает в себя вывод полученных результатов, построение графиков, проверку качества подгонки и диагностику сходимости.

Визуализация данных – **matplotlib** и **seaborn**

Python предлагает богатый выбор фреймворков для визуализации данных. Два самых популярных из них – **matplotlib** и более новый **seaborn**.

Matplotlib

Matplotlib – это библиотека Python для построения 2D-графиков, которая позволяет создавать высококачественные диаграммы в различных форматах и интерактивные окружения на разных платформах. Matplotlib можно использовать в скриптах Python, оболочке Python и IPython, Jupyter Notebook, серверах веб-приложений и еще в четырех инструментах, предлагающих графический пользовательский интерфейс.

Библиотека pandas очень тесно интегрирована с библиотекой matplotlib, включая методы объектов Series и DataFrame, которые позволяют автоматически вызывать библиотеку matplotlib. Это не означает, что возможности визуализации в библиотеке pandas ограничиваются только библиотекой matplotlib. Как мы увидим, можно легко воспользоваться и другими библиотеками, например ggplot2 и seaborn.

Seaborn

Seaborn (<http://seaborn.pydata.org/introduction.html>) – это библиотека для создания привлекательных и информативных статистических графиков в среде Python. Она надстраивается поверх библиотеки matplotlib и тесно интегрирована со стеком PyData, включая поддержку NumPy, объектов pandas и статистических процедур SciPy и StatsModels. Она обеспечивает дополнительную функциональность, выходящую за пределы библиотеки matplotlib, а также по умолчанию демонстрирует более богатые и более современные способы визуализации, чем matplotlib.

Выводы

В этой главе мы продолжили наше знакомство с библиотекой pandas, обработкой данных/анализом и наукой. Оно началось с обсуждения причин появления библиотеки pandas, ее функционала и ее взаимосвязи с такими понятиями, как обработка данных, анализ и наука о данных.

Затем мы рассмотрели процесс анализа данных, чтобы понять предназначение некоторых функций библиотеки pandas. Они включают в себя сбор, организацию, чистку и исследование данных, а затем построение формальной модели, представление результатов, а также возможность совместного использования и воспроизведения результатов анализа.

Затем мы рассмотрели несколько понятий, связанных с данными и статистическим моделированием. Мы рассмотрели общие методы анализа и понятия, чтобы познакомить вас с ними перед их подробным изучением в последующих главах.

Кроме того, `pandas` является частью более обширной экосистемы Python, предлагающей массу библиотек, полезных для анализа данных и науки. Хотя эта книга будет сосредоточена только на библиотеке `pandas`, есть и другие библиотеки, с которыми вы столкнетесь, и поэтому мы рассказали о них для общего знакомства.

Итак, мы готовы к использованию библиотеки `pandas`. В следующей главе, прежде чем углубляться в `pandas`, мы разберем ее основы, начав с окружения Python и `pandas`, обзора тетрадок Jupyter и закончив кратким знакомством с объектами библиотеки `pandas Series` и `DataFrame`.