Mastering pandas

Python is a groundbreaking language for its simplicity and succinctness, allowing the user to achieve a great deal with a few lines of code, especially compared to other programming languages. The pandas brings these features of Python into the data analysis realm, by providing expressiveness, simplicity, and powerful capabilities for the task of data analysis. By mastering pandas, users will be able to do complex data analysis in a short period of time, as well as illustrate their findings using the rich visualization capabilities of related tools such as IPython and matplotlib.

This book is an in-depth guide to the use of pandas for data analysis, for either the seasoned data analysis practitioner or the novice user. It provides a basic introduction to the pandas framework, and takes users through the installation of the library and the IPython interactive environment. Thereafter, you will learn basic as well as advanced features, such as MultiIndexing, modifying data structures, and sampling data, which provide powerful capabilities for data analysis.

Who this book is written for

This book is intended for Python programmers, mathematicians, and analysts who already have a basic understanding of Python and wish to learn about its data analysis capabilities in depth.

What you will learn from this book

- Download, install, and set up Python, pandas, and related tools to perform data analysis for different operating environments
- Practice using IPython as an interactive environment for doing data analysis using pandas
- Master the core features of pandas used in data analysis
- Get to grips with the more advanced features of pandas
- Understand the basics of using matplotlib to plot data analysis results
- Analyze real-world datasets using pandas
- Acquire knowledge of using pandas for basic statistical analysis

In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 'Introduction to pandas and Data Analysis'
- A synopsis of the book’s content
- More information on Mastering pandas
About the Author

**Femi Anthony** is a seasoned and knowledgeable software programmer, with over 15 years experience in a vast array of languages, including Perl, C, C++, Java, and Python. He has worked in both the Internet space and financial services space for many years and is now working for a well-known financial data company. He holds a bachelor's degree in mathematics with computer science from MIT and a master's degree from the University of Pennsylvania. His pet interests include data science, machine learning, and Python. Femi is working on a few side projects in these areas. His hobbies include reading, soccer, and road cycling. You can follow him at @dataphanatik, and for any queries, contact him at femibyte@gmail.com.
Preface

Welcome to Mastering pandas. This book will teach you how to effectively use pandas, which is one of the most popular Python packages today for performing data analysis. The first half of this book starts off with the rationale for performing data analysis. Then it introduces Python and pandas in particular, taking you through the installation steps, what pandas is all about, what it can be used for, data structures in pandas, and how to select, merge and group data in pandas. Then it covers handling missing data and time series data, as well as plotting for data visualization.

The second half of this book shows you how to use pandas to perform inferential statistics using the classical and Bayesian approaches, followed by a chapter on pandas architecture, before rounding off with a whirlwind tour of machine learning, which introduces the scikit-learn library. The aim of this book is to immerse you into pandas through the use of illustrative examples on real-world datasets.

What this book covers

Chapter 1, Introduction to pandas and Data Analysis, explains the motivation for doing data analysis, introduces the Python language and the pandas library, and discusses how they can be used for data analysis. It also describes the benefits of using pandas for data analysis.

Chapter 2, Installation of pandas and the Supporting Software, gives a detailed description on how to install pandas. It gives installation instructions across multiple operating system platforms: Unix, MacOS X, and Windows. It also describes how to install supporting software, such as NumPy and IPython.
Chapter 3, *The pandas Data Structures*, introduces the data structures that form the bedrock of the pandas library. The `numpy.ndarray` data structure is first introduced and discussed as it forms the basis for the `pandas.Series` and `pandas.DataFrame` data structures, which are the foundation data structures used in pandas. This chapter may be the most important one in the book, as knowledge of these data structures is absolutely necessary to do data analysis using pandas.

Chapter 4, *Operations in pandas, Part I – Indexing and Selecting*, focuses on how to access and select data from the pandas data structures. It discusses the various ways of selecting data via Basic, Label, Integer, and Mixed Indexing. It explains more advanced indexing concepts such as MultiIndex, Boolean indexing, and operations on Index types.

Chapter 5, *Operations in pandas, Part II – Grouping, Merging, and Reshaping of Data*, tackles the problem of rearranging data in pandas’ data structures. The various functions in pandas that enable the user to rearrange data are examined by utilizing them on real-world datasets. This chapter examines the different ways in which data can be rearranged: by aggregation/grouping, merging, concatenating, and reshaping.

Chapter 6, *Missing Data, Time Series, and Plotting using Matplotlib*, discusses topics that are necessary for the pre-processing of data that is to be used as input for data analysis, prediction, and visualization. These topics include how to handle missing values in the input data, how to handle time series data, and how to use the matplotlib library to plot data for visualization purposes.

Chapter 7, *A Tour of Statistics – The Classical Approach*, takes you on a brief tour of classical statistics and shows how pandas can be used together with Python’s statistical packages to conduct statistical analyses. Various statistical topics are addressed, including statistical inference, measures of central tendency, hypothesis testing, Z- and T-tests, analysis of variance, confidence intervals, and correlation and regression.

Chapter 8, *A Brief Tour of Bayesian Statistics*, discusses an alternative approach to performing statistical analysis, known as Bayesian analysis. This chapter introduces Bayesian statistics and discusses the underlying mathematical framework. It examines the various probability distributions used in Bayesian analysis and shows how to generate and visualize them using matplotlib and scipy.stats. It also introduces the PyMC library for performing Monte Carlo simulations, and provides a real-world example of conducting a Bayesian inference using online data.

Chapter 9, *The pandas Library Architecture*, provides a fairly detailed description of the code underlying pandas. It gives a breakdown of how the pandas library code is organized and describes the various modules that make up pandas, with some details. It also has a section that shows the user how to improve Python and pandas’s performance using extensions.
Chapter 10, *R and pandas Compared*, focuses on comparing pandas with R, the stats package on which much of pandas’s functionality is based. This chapter compares R data types and their pandas equivalents, and shows how the various operations compare in both libraries. Operations such as slicing, selection, arithmetic operations, aggregation, group-by, matching, split-apply-combine, and melting are compared.

Chapter 11, *Brief Tour of Machine Learning*, takes you on a whirlwind tour of machine learning, with focus on using the pandas library as a tool to preprocess input data into machine learning programs. It also introduces the scikit-learn library, which is the most widely used machine learning toolkit in Python. Various machine learning techniques and algorithms are introduced by applying them to a well-known machine learning classification problem: which passengers survived the sinking of the Titanic?
Introduction to pandas and Data Analysis

In this chapter, we address the following:

• Motivation for data analysis
• How Python and pandas can be used for data analysis
• Description of the pandas library
• Benefits of using pandas

Motivation for data analysis
In this section, we will discuss the trends that are making data analysis an increasingly important field of endeavor in today's fast-moving technological landscape.

We live in a big data world
The term big data has become one of the hottest technology buzzwords in the past two years. We now increasingly hear about big data in various media outlets, and big data startup companies have increasingly been attracting venture capital. A good example in the area of retail would be Target Corporation, which has invested substantially in big data and is now able to identify potential customers by using big data to analyze people's shopping habits online; refer to a related article at http://nyti.ms/19LT8ic.

Loosely speaking, big data refers to the phenomenon wherein the amount of data exceeds the capability of the recipients of the data to process it. Here is a Wikipedia entry on big data that sums it up nicely: http://en.wikipedia.org/wiki/Big_data.
4 V's of big data

A good way to start thinking about the complexities of big data is along what are called the 4 dimensions, or 4 V's of big data. This model was first introduced as the 3V's by Gartner analyst Doug Laney in 2001. The 3V's stood for Volume, Velocity, and Variety, and the 4th V, Veracity, was added later by IBM. Gartner's official definition is as follows:

"Big data is high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization."

Laney, Douglas. "The Importance of 'Big Data': A Definition", Gartner

Volume of big data

The volume of data in the big data age is simply mind-boggling. According to IBM, by 2020, the total amount of data on the planet would have ballooned to 40 zettabytes. You heard that right-40 zettabytes is 43 trillion gigabytes, which is about $4 \times 10^{21}$ bytes. For more information on this refer to the Wikipedia page on Zettabyte - http://en.wikipedia.org/wiki/Zettabyte.

To get a handle of how much data this would be, let me refer to an EMC press release published in 2010, which stated what 1 zettabyte was approximately equal to:

"The digital information created by every man, woman and child on Earth 'Tweeting' continuously for 100 years " or "75 billion fully-loaded 16 GB Apple iPads, which would fill the entire area of Wembley Stadium to the brim 41 times, the Mont Blanc Tunnel 84 times, CERN's Large Hadron Collider tunnel 151 times, Beijing National Stadium 15.5 times or the Taipei 101 Tower 23 times..."

EMC study projects 45× data growth by 2020

The growth rate of data has been fuelled largely by a few factors, such as the following:

- The rapid growth of the Internet.
- The conversion from analog to digital media coupled with an increased capability to capture and store data, which in turn has been made possible with cheaper and more capable storage technology. There has been a proliferation of digital data input devices such as cameras and wearables, and the cost of huge data storage has fallen rapidly. Amazon Web Services is a prime example of the trend toward much cheaper storage.
The Internetification of devices, or rather Internet of Things, is the phenomenon wherein common household devices, such as our refrigerators and cars, will be connected to the Internet. This phenomenon will only accelerate the above trend.

**Velocity of big data**

From a purely technological point of view, velocity refers to the throughput of big data, or how fast the data is coming in and is being processed. This has ramifications on how fast the recipient of the data needs to process it to keep up. Real-time analytics is one attempt to handle this characteristic. Tools that can help enable this include Amazon Web Services Elastic Map Reduce.

At a more macro level, the velocity of data can also be regarded as the increased speed at which data and information can now be transferred and processed faster and at greater distances than ever before.

The proliferation of high-speed data and communication networks coupled with the advent of cell phones, tablets, and other connected devices, are primary factors driving information velocity. Some measures of velocity include the number of tweets per second and the number of emails per minute.

**Variety of big data**

The variety of big data comes from having a multiplicity of data sources that generate the data, and the different formats of the data that are produced.

This results in a technological challenge for the recipients of the data who have to process it. Digital cameras, sensors, the web, cell phones, and so on are some of the data generators that produce data in differing formats, and the challenge comes in being able to handle all these formats and extract meaningful information from the data. The ever-changing nature of the data formats with the dawn of the big data era has led to a revolution in the database technology industry, with the rise of NoSQL databases to handle what is known as unstructured data or rather data whose format is fungible or constantly changing. For more information on Couchbase, refer to "Why NoSQL- http://bit.ly/1c3iVEc."
Veracity of big data

The 4th characteristic of big data – veracity, which was added later, refers to the need to validate or confirm the correctness of the data or the fact that the data represents the truth. The sources of data must be verified and the errors kept to a minimum. According to an estimate by IBM, poor data quality costs the US economy about $3.1 trillion dollars a year. For example, medical errors cost the United States $19.5 billion in 2008; for more information you can refer to a related article at http://bit.ly/1CTah5r. Here is an info-graphic by IBM that summarizes the 4V's of big data:

![IBM on the 4 V's of big data](image)

So much data, so little time for analysis

Data analytics has been described by Eric Schmidt, the former CEO of Google, as the Future of Everything. For reference, you can check out a YouTube video called Why Data Analytics is the Future of Everything at http://bit.ly/1KmgGCP.
The volume and velocity of data will continue to increase in the big data age. Companies that can efficiently collect, filter, and analyze data results in information that allows them to better meet the needs of their customers in a much quicker timeframe will gain a significant competitive advantage over their competitors. For example, data analytics (Culture of Metrics) plays a very key role in the business strategy of http://www.amazon.com/. For more information refer to Amazon.com Case Study, Smart Insights at http://bit.ly/1g1nAlu.

**The move towards real-time analytics**

As technologies and tools have evolved, to meet the ever-increasing demands of business, there has been a move towards what is known as real-time analytics. More information on Insight Everywhere, Intel available at http://intel.ly/1899xqo.

In the big data Internet era, here are some examples:

- Online businesses demand instantaneous insights into how the new products/features they have introduced in their online market are doing and how they can adjust their online product mix accordingly. Amazon is a prime example of this with their Customers Who Viewed This Item Also Viewed feature.

- In finance, risk management and trading systems demand almost instantaneous analysis in order to make effective decisions based on data-driven insights.

**How Python and pandas fit into the data analytics mix**

The Python programming language is one of the fastest growing languages today in the emerging field of data science and analytics. Python was created by Guido von Russom in 1991, and its key features include the following:

- Interpreted rather than compiled
- Dynamic type system
- Pass by value with object references
- Modular capability
- Comprehensive libraries
- Extensibility with respect to other languages
• Object orientation
• Most of the major programming paradigms-procedural, object-oriented, and to a lesser extent, functional.

Among the characteristics that make Python popular for data science are its very user-friendly (human-readable) syntax, the fact that it is interpreted rather than compiled (leading to faster development time), and its very comprehensive library for parsing and analyzing data, as well as its capacity for doing numerical and statistical computations. Python has libraries that provide a complete toolkit for data science and analysis. The major ones are as follows:

• NumPy: The general-purpose array functionality with emphasis on numeric computation
• SciPy: Numerical computing
• Matplotlib: Graphics
• pandas: Series and data frames (1D and 2D array-like types)
• Scikit-Learn: Machine learning
• NLTK: Natural language processing
• Statstool: Statistical analysis

For this book, we will be focusing on the 4th library listed in the preceding list, pandas.

What is pandas?
The pandas is a high-performance open source library for data analysis in Python developed by Wes McKinney in 2008. Over the years, it has become the de-facto standard library for data analysis using Python. There's been great adoption of the tool, a large community behind it, (220+ contributors and 9000+ commits by 03/2014), rapid iteration, features, and enhancements continuously made.

Some key features of pandas include the following:

• It can process a variety of data sets in different formats: time series, tabular heterogeneous, and matrix data.
• It facilitates loading/importing data from varied sources such as CSV and DB/SQL.
- It can handle a myriad of operations on data sets: subsetting, slicing, filtering, merging, groupBy, re-ordering, and re-shaping.
- It can deal with missing data according to rules defined by the user/developer: ignore, convert to 0, and so on.
- It can be used for parsing and munging (conversion) of data as well as modeling and statistical analysis.
- It integrates well with other Python libraries such as statsmodels, SciPy, and scikit-learn.
- It delivers fast performance and can be speeded up even more by making use of Cython (C extensions to Python).

For more information go through the official pandas documentation available at http://pandas.pydata.org/pandas-docs/stable/.

Benefits of using pandas

The pandas forms a core component of the Python data analysis corpus. The distinguishing feature of pandas is the suite of data structures that it provides, which is naturally suited to data analysis, primarily the DataFrame and to a lesser extent Series (1-D vectors) and Panel (3D tables).

Simply put, pandas and statstools can be described as Python's answer to R, the data analysis and statistical programming language that provides both the data structures, such as R-data frames, and a rich statistical library for data analysis.

The benefits of pandas over using a language such as Java, C, or C++ for data analysis are manifold:

- **Data representation**: It can easily represent data in a form naturally suited for data analysis via its DataFrame and Series data structures in a concise manner. Doing the equivalent in Java/C/C++ would require many lines of custom code, as these languages were not built for data analysis but rather networking and kernel development.
- **Data subsetting and filtering**: It provides for easy subsetting and filtering of data, procedures that are a staple of doing data analysis.
Concise and clear code: Its concise and clear API allows the user to focus more on the core goal at hand, rather than have to write a lot of scaffolding code in order to perform routine tasks. For example, reading a CSV file into a DataFrame data structure in memory takes two lines of code, while doing the same task in Java/C/C++ would require many more lines of code or calls to non-standard libraries, as illustrated in the following table. Here, let's suppose that we had the following data:

<table>
<thead>
<tr>
<th>Country</th>
<th>Year</th>
<th>CO2 Emissions</th>
<th>Power Consumption</th>
<th>Fertility Rate</th>
<th>Internet Usage Per 100 People</th>
<th>Life Expectancy</th>
<th>Population</th>
</tr>
</thead>
<tbody>
<tr>
<td>Belarus</td>
<td>2000</td>
<td>5.91</td>
<td>2988.71</td>
<td>1.29</td>
<td>18.69</td>
<td>68.01</td>
<td>1.00E+07</td>
</tr>
<tr>
<td>Belarus</td>
<td>2001</td>
<td>5.87</td>
<td>2996.81</td>
<td></td>
<td>43.15</td>
<td></td>
<td>9970260</td>
</tr>
<tr>
<td>Belarus</td>
<td>2002</td>
<td>6.03</td>
<td>2982.77</td>
<td>1.25</td>
<td>89.8</td>
<td>68.21</td>
<td>9925000</td>
</tr>
<tr>
<td>Belarus</td>
<td>2003</td>
<td>6.33</td>
<td>3039.1</td>
<td>1.25</td>
<td>162.76</td>
<td></td>
<td>9873968</td>
</tr>
<tr>
<td>Belarus</td>
<td>2004</td>
<td></td>
<td>3143.58</td>
<td>1.24</td>
<td>250.51</td>
<td>68.39</td>
<td>9824469</td>
</tr>
<tr>
<td>Belarus</td>
<td>2005</td>
<td></td>
<td></td>
<td>1.24</td>
<td>347.23</td>
<td>68.48</td>
<td>9775591</td>
</tr>
</tbody>
</table>

In a CSV file, this data that we wish to read would look like the following:

```
Country,Year,CO2Emissions,PowerConsumption,FertilityRate,
InternetUsagePer10000,LifeExpectancy,Population
Belarus,2000,5.91,2988.71,1.29,18.69,68.01,1.00E+07
Belarus,2001,5.87,2996.81,,43.15,,9970260
Belarus,2002,6.03,2982.77,1.25,89.8,68.21,9925000
... 
Philippines,2000,1.03,514.02,,20.33,69.53,7.58E+07
Philippines,2001,0.99,535.18,,25.89,,7.72E+07
Philippines,2002,0.99,539.74,3.5,44.47,70.19,7.87E+07
... 
Morocco,2000,1.2,489.04,2.62,7.03,68.81,2.85E+07
Morocco,2001,1.32,508.1,2.5,13.87,,2.88E+07
Morocco,2002,1.32,526.4,2.5,23.99,69.48,2.92E+07
...
```


In Java, we would have to write the following code:

```java
public class CSVReader {
    public static void main(String[] args) {
```
String[] csvFile=argv[1];
CSVReader csvReader = new csvReader();
List<Map> dataTable=csvReader.readCSV(csvFile);
}

public void readCSV(String[] csvFile)
{
    BufferedReader bReader=null;
    String line="";
    String delim="",";
    //Initialize List of maps, each representing a line of the csv file
    List<Map> data=new ArrayList<Map>();
    try {
        bufferedReader = new BufferedReader(new
        FileReader(csvFile));
        // Read the csv file, line by line
        while ((line = br.readLine()) != null){
            String[] row = line.split(delim);
            Map<String,String> csvRow=new HashMap<String,String>();
            csvRow.put('Country')=row[0];
            csvRow.put('Year')=row[1];
            csvRow.put('CO2Emissions')=row[2];
            csvRow.put('PowerConsumption')=row[3];
            csvRow.put('FertilityRate')=row[4];
            csvRow.put('InternetUsage')=row[5];
            csvRow.put('LifeExpectancy')=row[6];
            csvRow.put('Population')=row[7];
            data.add(csvRow);
        }
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return data;
}

But, using pandas, it would take just two lines of code:

    import pandas as pd
    worldBankDF=pd.read_csv('worldbank.csv')

In addition, pandas is built upon the NumPy libraries and hence, inherits many of
the performance benefits of this package, especially when it comes to numerical and
scientific computing. One oft-touted drawback of using Python is that as a scripting
language, its performance relative to languages like Java/C/C++ has been rather
slow. However, this is not really the case for pandas.
Summary

We live in a big data era characterized by the 4V's- volume, velocity, variety, and veracity. The volume and velocity of data are ever increasing for the foreseeable future. Companies that can harness and analyze big data to extract information and take actionable decisions based on this information will be the winners in the marketplace. Python is a fast-growing, user-friendly, extensible language that is very popular for data analysis.

The pandas is a core library of the Python toolkit for data analysis. It provides features and capabilities that make it much easier and faster for data analysis than many other popular languages such as Java, C, C++, and Ruby.

Thus, given the strengths of Python listed in the preceding section as a choice for the analysis of data, the data analysis practitioner utilizing Python should become quite adept at pandas in order to become more effective. This book aims to assist the user in achieving this goal.
Where to buy this book

You can buy Mastering pandas from the Packt Publishing website.
Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.