

O'REILLY®

2nd Edition

Python for Data Analysis

DATA WRANGLING WITH PANDAS,
NUMPY, AND IPYTHON



powered by



Wes McKinney

SECOND EDITION

Python for Data Analysis

*Data Wrangling with Pandas, NumPy,
and IPython*

Wes McKinney

Beijing • Boston • Farnham • Sebastopol • Tokyo

O'REILLY®

Python for Data Analysis

by Wes McKinney

Copyright © 2018 William McKinney. All rights reserved.

Printed in the United States of America.

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

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://oreilly.com/safari>). For more information, contact our corporate/institutional sales department: 800-998-9938 or corporate@oreilly.com.

Editor: Marie Beaugureau

Production Editor: Kristen Brown

Copyeditor: Jasmine Kwityn

Proofreader: Rachel Monaghan

Indexer: Lucie Haskins

Interior Designer: David Futato

Cover Designer: Karen Montgomery

Illustrator: Rebecca Demarest

October 2012: First Edition

October 2017: Second Edition

Revision History for the Second Edition

2017-09-25: First Release

See <http://oreilly.com/catalog/errata.csp?isbn=9781491957660> for release details.

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. *Python for Data Analysis*, the cover image, and related trade dress are trademarks of O'Reilly Media, Inc.

While the publisher and the author have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the author disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

978-1-491-95766-0

[LSI]

Table of Contents

Preface	xi
1. Preliminaries	1
1.1 What Is This Book About?	1
What Kinds of Data?	1
1.2 Why Python for Data Analysis?	2
Python as Glue	2
Solving the “Two-Language” Problem	3
Why Not Python?	3
1.3 Essential Python Libraries	4
NumPy	4
pandas	4
matplotlib	5
IPython and Jupyter	6
SciPy	6
scikit-learn	7
statsmodels	8
1.4 Installation and Setup	8
Windows	9
Apple (OS X, macOS)	9
GNU/Linux	9
Installing or Updating Python Packages	10
Python 2 and Python 3	11
Integrated Development Environments (IDEs) and Text Editors	11
1.5 Community and Conferences	12
1.6 Navigating This Book	12
Code Examples	13
Data for Examples	13

Import Conventions	14
Jargon	14
2. Python Language Basics, IPython, and Jupyter Notebooks.	15
2.1 The Python Interpreter	16
2.2 IPython Basics	17
Running the IPython Shell	17
Running the Jupyter Notebook	18
Tab Completion	21
Introspection	23
The %run Command	25
Executing Code from the Clipboard	26
Terminal Keyboard Shortcuts	27
About Magic Commands	28
Matplotlib Integration	29
2.3 Python Language Basics	30
Language Semantics	30
Scalar Types	38
Control Flow	46
3. Built-in Data Structures, Functions, and Files.	51
3.1 Data Structures and Sequences	51
Tuple	51
List	54
Built-in Sequence Functions	59
dict	61
set	65
List, Set, and Dict Comprehensions	67
3.2 Functions	69
Namespaces, Scope, and Local Functions	70
Returning Multiple Values	71
Functions Are Objects	72
Anonymous (Lambda) Functions	73
Currying: Partial Argument Application	74
Generators	75
Errors and Exception Handling	77
3.3 Files and the Operating System	80
Bytes and Unicode with Files	83
3.4 Conclusion	84
4. NumPy Basics: Arrays and Vectorized Computation.	85
4.1 The NumPy ndarray: A Multidimensional Array Object	87

Creating ndarrays	88
Data Types for ndarrays	90
Arithmetic with NumPy Arrays	93
Basic Indexing and Slicing	94
Boolean Indexing	99
Fancy Indexing	102
Transposing Arrays and Swapping Axes	103
4.2 Universal Functions: Fast Element-Wise Array Functions	105
4.3 Array-Oriented Programming with Arrays	108
Expressing Conditional Logic as Array Operations	109
Mathematical and Statistical Methods	111
Methods for Boolean Arrays	113
Sorting	113
Unique and Other Set Logic	114
4.4 File Input and Output with Arrays	115
4.5 Linear Algebra	116
4.6 Pseudorandom Number Generation	118
4.7 Example: Random Walks	119
Simulating Many Random Walks at Once	121
4.8 Conclusion	122
5. Getting Started with pandas.....	123
5.1 Introduction to pandas Data Structures	124
Series	124
DataFrame	128
Index Objects	134
5.2 Essential Functionality	136
Reindexing	136
Dropping Entries from an Axis	138
Indexing, Selection, and Filtering	140
Integer Indexes	145
Arithmetic and Data Alignment	146
Function Application and Mapping	151
Sorting and Ranking	153
Axis Indexes with Duplicate Labels	157
5.3 Summarizing and Computing Descriptive Statistics	158
Correlation and Covariance	160
Unique Values, Value Counts, and Membership	162
5.4 Conclusion	165
6. Data Loading, Storage, and File Formats.....	167
6.1 Reading and Writing Data in Text Format	167

Reading Text Files in Pieces	173
Writing Data to Text Format	175
Working with Delimited Formats	176
JSON Data	178
XML and HTML: Web Scraping	180
6.2 Binary Data Formats	183
Using HDF5 Format	184
Reading Microsoft Excel Files	186
6.3 Interacting with Web APIs	187
6.4 Interacting with Databases	188
6.5 Conclusion	190
7. Data Cleaning and Preparation.....	191
7.1 Handling Missing Data	191
Filtering Out Missing Data	193
Filling In Missing Data	195
7.2 Data Transformation	197
Removing Duplicates	197
Transforming Data Using a Function or Mapping	198
Replacing Values	200
Renaming Axis Indexes	201
Discretization and Binning	203
Detecting and Filtering Outliers	205
Permutation and Random Sampling	206
Computing Indicator/Dummy Variables	208
7.3 String Manipulation	211
String Object Methods	211
Regular Expressions	213
Vectorized String Functions in pandas	216
7.4 Conclusion	219
8. Data Wrangling: Join, Combine, and Reshape.....	221
8.1 Hierarchical Indexing	221
Reordering and Sorting Levels	224
Summary Statistics by Level	225
Indexing with a DataFrame's columns	225
8.2 Combining and Merging Datasets	227
Database-Style DataFrame Joins	227
Merging on Index	232
Concatenating Along an Axis	236
Combining Data with Overlap	241
8.3 Reshaping and Pivoting	242

Reshaping with Hierarchical Indexing	243
Pivoting “Long” to “Wide” Format	246
Pivoting “Wide” to “Long” Format	249
8.4 Conclusion	251
9. Plotting and Visualization.....	253
9.1 A Brief matplotlib API Primer	253
Figures and Subplots	255
Colors, Markers, and Line Styles	259
Ticks, Labels, and Legends	261
Annotations and Drawing on a Subplot	265
Saving Plots to File	267
matplotlib Configuration	268
9.2 Plotting with pandas and seaborn	268
Line Plots	269
Bar Plots	272
Histograms and Density Plots	277
Scatter or Point Plots	280
Facet Grids and Categorical Data	283
9.3 Other Python Visualization Tools	285
9.4 Conclusion	286
10. Data Aggregation and Group Operations.....	287
10.1 GroupBy Mechanics	288
Iterating Over Groups	291
Selecting a Column or Subset of Columns	293
Grouping with Dicts and Series	294
Grouping with Functions	295
Grouping by Index Levels	295
10.2 Data Aggregation	296
Column-Wise and Multiple Function Application	298
Returning Aggregated Data Without Row Indexes	301
10.3 Apply: General split-apply-combine	302
Suppressing the Group Keys	304
Quantile and Bucket Analysis	305
Example: Filling Missing Values with Group-Specific Values	306
Example: Random Sampling and Permutation	308
Example: Group Weighted Average and Correlation	310
Example: Group-Wise Linear Regression	312
10.4 Pivot Tables and Cross-Tabulation	313
Cross-Tabulations: Crosstab	315
10.5 Conclusion	316

11. Time Series	317
11.1 Date and Time Data Types and Tools	318
Converting Between String and Datetime	319
11.2 Time Series Basics	322
Indexing, Selection, Subsetting	323
Time Series with Duplicate Indices	326
11.3 Date Ranges, Frequencies, and Shifting	327
Generating Date Ranges	328
Frequencies and Date Offsets	330
Shifting (Leading and Lagging) Data	332
11.4 Time Zone Handling	335
Time Zone Localization and Conversion	335
Operations with Time Zone–Aware Timestamp Objects	338
Operations Between Different Time Zones	339
11.5 Periods and Period Arithmetic	339
Period Frequency Conversion	340
Quarterly Period Frequencies	342
Converting Timestamps to Periods (and Back)	344
Creating a PeriodIndex from Arrays	345
11.6 Resampling and Frequency Conversion	348
Downsampling	349
Upsampling and Interpolation	352
Resampling with Periods	353
11.7 Moving Window Functions	354
Exponentially Weighted Functions	358
Binary Moving Window Functions	359
User-Defined Moving Window Functions	361
11.8 Conclusion	362
12. Advanced pandas	363
12.1 Categorical Data	363
Background and Motivation	363
Categorical Type in pandas	365
Computations with Categoricals	367
Categorical Methods	370
12.2 Advanced GroupBy Use	373
Group Transforms and “Unwrapped” GroupBys	373
Grouped Time Resampling	377
12.3 Techniques for Method Chaining	378
The pipe Method	380
12.4 Conclusion	381

13. Introduction to Modeling Libraries in Python.....	383
13.1 Interfacing Between pandas and Model Code	383
13.2 Creating Model Descriptions with Patsy	386
Data Transformations in Patsy Formulas	389
Categorical Data and Patsy	390
13.3 Introduction to statsmodels	393
Estimating Linear Models	393
Estimating Time Series Processes	396
13.4 Introduction to scikit-learn	397
13.5 Continuing Your Education	401
14. Data Analysis Examples.....	403
14.1 1.USA.gov Data from Bitly	403
Counting Time Zones in Pure Python	404
Counting Time Zones with pandas	406
14.2 MovieLens 1M Dataset	413
Measuring Rating Disagreement	418
14.3 US Baby Names 1880–2010	419
Analyzing Naming Trends	425
14.4 USDA Food Database	434
14.5 2012 Federal Election Commission Database	440
Donation Statistics by Occupation and Employer	442
Bucketing Donation Amounts	445
Donation Statistics by State	447
14.6 Conclusion	448
A. Advanced NumPy.....	449
A.1 ndarray Object Internals	449
NumPy dtype Hierarchy	450
A.2 Advanced Array Manipulation	451
Reshaping Arrays	452
C Versus Fortran Order	454
Concatenating and Splitting Arrays	454
Repeating Elements: tile and repeat	457
Fancy Indexing Equivalents: take and put	459
A.3 Broadcasting	460
Broadcasting Over Other Axes	462
Setting Array Values by Broadcasting	465
A.4 Advanced ufunc Usage	466
ufunc Instance Methods	466
Writing New ufuncs in Python	468
A.5 Structured and Record Arrays	469

Nested dtypes and Multidimensional Fields	469
Why Use Structured Arrays?	470
A.6 More About Sorting	471
Indirect Sorts: argsort and lexsort	472
Alternative Sort Algorithms	474
Partially Sorting Arrays	474
numpy.searchsorted: Finding Elements in a Sorted Array	475
A.7 Writing Fast NumPy Functions with Numba	476
Creating Custom numpy.ufunc Objects with Numba	478
A.8 Advanced Array Input and Output	478
Memory-Mapped Files	478
HDF5 and Other Array Storage Options	480
A.9 Performance Tips	480
The Importance of Contiguous Memory	480
B. More on the IPython System.....	483
B.1 Using the Command History	483
Searching and Reusing the Command History	483
Input and Output Variables	484
B.2 Interacting with the Operating System	485
Shell Commands and Aliases	486
Directory Bookmark System	487
B.3 Software Development Tools	487
Interactive Debugger	488
Timing Code: %time and %timeit	492
Basic Profiling: %prun and %run -p	494
Profiling a Function Line by Line	496
B.4 Tips for Productive Code Development Using IPython	498
Reloading Module Dependencies	498
Code Design Tips	499
B.5 Advanced IPython Features	500
Making Your Own Classes IPython-Friendly	500
Profiles and Configuration	501
B.6 Conclusion	503
Index.....	505

New for the Second Edition

The first edition of this book was published in 2012, during a time when open source data analysis libraries for Python (such as pandas) were very new and developing rapidly. In this updated and expanded second edition, I have overhauled the chapters to account both for incompatible changes and deprecations as well as new features that have occurred in the last five years. I've also added fresh content to introduce tools that either did not exist in 2012 or had not matured enough to make the first cut. Finally, I have tried to avoid writing about new or cutting-edge open source projects that may not have had a chance to mature. I would like readers of this edition to find that the content is still almost as relevant in 2020 or 2021 as it is in 2017.

The major updates in this second edition include:

- All code, including the Python tutorial, updated for Python 3.6 (the first edition used Python 2.7)
- Updated Python installation instructions for the Anaconda Python Distribution and other needed Python packages
- Updates for the latest versions of the pandas library in 2017
- A new chapter on some more advanced pandas tools, and some other usage tips
- A brief introduction to using statsmodels and scikit-learn

I also reorganized a significant portion of the content from the first edition to make the book more accessible to newcomers.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, and file extensions.

Constant width

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

Constant width bold

Shows commands or other text that should be typed literally by the user.

Constant width italic

Shows text that should be replaced with user-supplied values or by values determined by context.



This element signifies a tip or suggestion.



This element signifies a general note.



This element indicates a warning or caution.

Using Code Examples

You can find data files and related material for each chapter is available in this book's GitHub repository at <http://github.com/wesm/pydata-book>.


This book is here to help you get your job done. In general, if example code is offered with this book, you may use it in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this

book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: “*Python for Data Analysis* by Wes McKinney (O'Reilly). Copyright 2017 Wes McKinney, 978-1-491-95766-0.”

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.

O'Reilly Safari

 Safari (formerly Safari Books Online) is a membership-based training and reference platform for enterprise, government, educators, and individuals.

Members have access to thousands of books, training videos, Learning Paths, interactive tutorials, and curated playlists from over 250 publishers, including O'Reilly Media, Harvard Business Review, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Adobe, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, and Course Technology, among others.

For more information, please visit <http://oreilly.com/safari>.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at http://bit.ly/python_data_analysis_2e.

To comment or ask technical questions about this book, send email to bookquestions@oreilly.com.

For more information about our books, courses, conferences, and news, see our website at <http://www.oreilly.com>.

Find us on Facebook: <http://facebook.com/oreilly>

Follow us on Twitter: <http://twitter.com/oreillymedia>

Watch us on YouTube: <http://www.youtube.com/oreillymedia>

Acknowledgments

This work is the product of many years of fruitful discussions, collaborations, and assistance with and from many people around the world. I'd like to thank a few of them.

In Memoriam: John D. Hunter (1968–2012)

Our dear friend and colleague John D. Hunter passed away after a battle with colon cancer on August 28, 2012. This was only a short time after I'd completed the final manuscript for this book's first edition.

John's impact and legacy in the Python scientific and data communities would be hard to overstate. In addition to developing matplotlib in the early 2000s (a time when Python was not nearly so popular), he helped shape the culture of a critical generation of open source developers who've become pillars of the Python ecosystem that we now often take for granted.

I was lucky enough to connect with John early in my open source career in January 2010, just after releasing pandas 0.1. His inspiration and mentorship helped me push forward, even in the darkest of times, with my vision for pandas and Python as a first-class data analysis language.

John was very close with Fernando Pérez and Brian Granger, pioneers of IPython, Jupyter, and many other initiatives in the Python community. We had hoped to work on a book together, the four of us, but I ended up being the one with the most free time. I am sure he would be proud of what we've accomplished, as individuals and as a community, over the last five years.

Acknowledgments for the Second Edition (2017)

It has been five years almost to the day since I completed the manuscript for this book's first edition in July 2012. A lot has changed. The Python community has grown immensely, and the ecosystem of open source software around it has flourished.

This new edition of the book would not exist if not for the tireless efforts of the pandas core developers, who have grown the project and its user community into one of the cornerstones of the Python data science ecosystem. These include, but are not limited to, Tom Augspurger, Joris van den Bossche, Chris Bartak, Phillip Cloud, gyoung, Andy Hayden, Masaaki Horikoshi, Stephan Hoyer, Adam Klein, Wouter Overmeire, Jeff Reback, Chang She, Skipper Seabold, Jeff Tratner, and y-p.

On the actual writing of this second edition, I would like to thank the O'Reilly staff who helped me patiently with the writing process. This includes Marie Beaugureau, Ben Lorica, and Colleen Toporek. I again had outstanding technical reviewers with Tom Augspurger, Paul Barry, Hugh Brown, Jonathan Coe, and Andreas Müller contributing. Thank you.

This book's first edition has been translated into many foreign languages, including Chinese, French, German, Japanese, Korean, and Russian. Translating all this content and making it available to a broader audience is a huge and often thankless effort. Thank you for helping more people in the world learn how to program and use data analysis tools.

I am also lucky to have had support for my continued open source development efforts from Cloudera and Two Sigma Investments over the last few years. With open source software projects more thinly resourced than ever relative to the size of user bases, it is becoming increasingly important for businesses to provide support for development of key open source projects. It's the right thing to do.

Acknowledgments for the First Edition (2012)

It would have been difficult for me to write this book without the support of a large number of people.

On the O'Reilly staff, I'm very grateful for my editors, Meghan Blanchette and Julie Steele, who guided me through the process. Mike Loukides also worked with me in the proposal stages and helped make the book a reality.

I received a wealth of technical review from a large cast of characters. In particular, Martin Blais and Hugh Brown were incredibly helpful in improving the book's examples, clarity, and organization from cover to cover. James Long, Drew Conway, Fernando Pérez, Brian Granger, Thomas Kluyver, Adam Klein, Josh Klein, Chang She, and Stéfan van der Walt each reviewed one or more chapters, providing pointed feedback from many different perspectives.

I got many great ideas for examples and datasets from friends and colleagues in the data community, among them: Mike Dewar, Jeff Hammerbacher, James Johndrow, Kristian Lum, Adam Klein, Hilary Mason, Chang She, and Ashley Williams.

I am of course indebted to the many leaders in the open source scientific Python community who've built the foundation for my development work and gave encouragement while I was writing this book: the IPython core team (Fernando Pérez, Brian Granger, Min Ragan-Kelly, Thomas Kluyver, and others), John Hunter, Skipper Seabold, Travis Oliphant, Peter Wang, Eric Jones, Robert Kern, Josef Perktold, Francesc Alted, Chris Fonnesbeck, and too many others to mention. Several other people provided a great deal of support, ideas, and encouragement along the way: Drew Conway, Sean Taylor, Giuseppe Paleologo, Jared Lander, David Epstein, John Krowas, Joshua Bloom, Den Pilsworth, John Myles-White, and many others I've forgotten.

I'd also like to thank a number of people from my formative years. First, my former AQR colleagues who've cheered me on in my pandas work over the years: Alex Reyman, Michael Wong, Tim Sargen, Oktay Kurbanov, Matthew Tschantz, Roni Israelov, Michael Katz, Chris Uga, Prasad Ramanan, Ted Square, and Hoon Kim. Lastly, my academic advisors Haynes Miller (MIT) and Mike West (Duke).

I received significant help from Phillip Cloud and Joris Van den Bossche in 2014 to update the book's code examples and fix some other inaccuracies due to changes in pandas.

On the personal side, Casey provided invaluable day-to-day support during the writing process, tolerating my highs and lows as I hacked together the final draft on top of an already overcommitted schedule. Lastly, my parents, Bill and Kim, taught me to always follow my dreams and to never settle for less.

Preliminaries

1.1 What Is This Book About?

This book is concerned with the nuts and bolts of manipulating, processing, cleaning, and crunching data in Python. My goal is to offer a guide to the parts of the Python programming language and its data-oriented library ecosystem and tools that will equip you to become an effective data analyst. While “data analysis” is in the title of the book, the focus is specifically on Python programming, libraries, and tools as opposed to data analysis methodology. This is the Python programming you need *for* data analysis.

What Kinds of Data?

When I say “data,” what am I referring to exactly? The primary focus is on *structured data*, a deliberately vague term that encompasses many different common forms of data, such as:

- Tabular or spreadsheet-like data in which each column may be a different type (string, numeric, date, or otherwise). This includes most kinds of data commonly stored in relational databases or tab- or comma-delimited text files.
- Multidimensional arrays (matrices).
- Multiple tables of data interrelated by key columns (what would be primary or foreign keys for a SQL user).
- Evenly or unevenly spaced time series.

This is by no means a complete list. Even though it may not always be obvious, a large percentage of datasets can be transformed into a structured form that is more suitable for analysis and modeling. If not, it may be possible to extract features from a dataset

into a structured form. As an example, a collection of news articles could be processed into a word frequency table, which could then be used to perform sentiment analysis.

Most users of spreadsheet programs like Microsoft Excel, perhaps the most widely used data analysis tool in the world, will not be strangers to these kinds of data.

1.2 Why Python for Data Analysis?

For many people, the Python programming language has strong appeal. Since its first appearance in 1991, Python has become one of the most popular interpreted programming languages, along with Perl, Ruby, and others. Python and Ruby have become especially popular since 2005 or so for building websites using their numerous web frameworks, like Rails (Ruby) and Django (Python). Such languages are often called *scripting* languages, as they can be used to quickly write small programs, or *scripts* to automate other tasks. I don't like the term "scripting language," as it carries a connotation that they cannot be used for building serious software. Among interpreted languages, for various historical and cultural reasons, Python has developed a large and active scientific computing and data analysis community. In the last 10 years, Python has gone from a bleeding-edge or "at your own risk" scientific computing language to one of the most important languages for data science, machine learning, and general software development in academia and industry.

For data analysis and interactive computing and data visualization, Python will inevitably draw comparisons with other open source and commercial programming languages and tools in wide use, such as R, MATLAB, SAS, Stata, and others. In recent years, Python's improved support for libraries (such as pandas and scikit-learn) has made it a popular choice for data analysis tasks. Combined with Python's overall strength for general-purpose software engineering, it is an excellent option as a primary language for building data applications.

Python as Glue

Part of Python's success in scientific computing is the ease of integrating C, C++, and FORTRAN code. Most modern computing environments share a similar set of legacy FORTRAN and C libraries for doing linear algebra, optimization, integration, fast Fourier transforms, and other such algorithms. The same story has held true for many companies and national labs that have used Python to glue together decades' worth of legacy software.

Many programs consist of small portions of code where most of the time is spent, with large amounts of "glue code" that doesn't run often. In many cases, the execution time of the glue code is insignificant; effort is most fruitfully invested in optimizing