



Python Data Science Handbook

ESSENTIAL TOOLS FOR WORKING WITH DATA

powered by



Jake VanderPlas

Python Data Science Handbook

For many researchers, Python is a first-class tool mainly because of its libraries for storing, manipulating, and gaining insight from data. Several resources exist for individual pieces of this data science stack, but only with the *Python Data Science Handbook* do you get them all—IPython, NumPy, Pandas, Matplotlib, Scikit-Learn, and other related tools.

Working scientists and data crunchers familiar with reading and writing Python code will find this comprehensive desk reference ideal for tackling day-to-day issues: manipulating, transforming, and cleaning data; visualizing different types of data; and using data to build statistical or machine learning models. Quite simply, this is the must-have reference for scientific computing in Python.

With this handbook, you'll learn how to use:

- **IPython and Jupyter:** provide computational environments for data scientists using Python
- **NumPy:** includes the *ndarray* for efficient storage and manipulation of dense data arrays in Python
- **Pandas:** features the *DataFrame* for efficient storage and manipulation of labeled/columnar data in Python
- **Matplotlib:** includes capabilities for a flexible range of data visualizations in Python
- **Scikit-Learn:** for efficient and clean Python implementations of the most important and established machine learning algorithms

“If you want to learn data science with Python, this book is a fantastic starting point. I've used it with great success to teach computer science and statistics majors. Jake goes far beyond the basics of open source tools; he also explains the underlying concepts, patterns, and abstractions of data science using clear language and approachable explanations.”

—**Brian Granger**

Associate Professor of Physics,
Cal Poly; cofounder of Project Jupyter

Jake VanderPlas, a long-time user and developer of the Python scientific stack, currently works as an interdisciplinary research director at the University of Washington. He conducts his own astronomy research, and spends time advising and consulting with local scientists from a wide range of fields.

PYTHON / DATA

US \$59.99

CAN \$68.99

ISBN: 978-1-491-91205-8



5 5 9 9 9

9 781491 912058



Twitter: @oreillymedia
facebook.com/oreilly

Python Data Science Handbook

Essential Tools for Working with Data

Jake VanderPlas

Beijing • Boston • Farnham • Sebastopol • Tokyo

O'REILLY®

Python Data Science Handbook

by Jake VanderPlas

Copyright © 2017 Jake VanderPlas. 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: Dawn Schanafelt

Indexer: WordCo Indexing Services, Inc.

Production Editor: Kristen Brown

Interior Designer: David Futato

Copyeditor: Jasmine Kwityn

Cover Designer: Karen Montgomery

Proofreader: Rachel Monaghan

Illustrator: Rebecca Demarest

December 2016: First Edition

Revision History for the First Edition

2016-11-17: First Release

See <http://oreilly.com/catalog/errata.csp?isbn=9781491912058> for release details.

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. *Python Data Science Handbook*, 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.

978-1-491-91205-8

[LSI]

Table of Contents

Preface.....	xı
1. IPython: Beyond Normal Python.....	1
Shell or Notebook?	2
Launching the IPython Shell	2
Launching the Jupyter Notebook	2
Help and Documentation in IPython	3
Accessing Documentation with ?	3
Accessing Source Code with ??	5
Exploring Modules with Tab Completion	6
Keyboard Shortcuts in the IPython Shell	8
Navigation Shortcuts	8
Text Entry Shortcuts	9
Command History Shortcuts	9
Miscellaneous Shortcuts	10
IPython Magic Commands	10
Pasting Code Blocks: %paste and %cpaste	11
Running External Code: %run	12
Timing Code Execution: %timeit	12
Help on Magic Functions: ?, %magic, and %lsmagic	13
Input and Output History	13
IPython's In and Out Objects	13
Underscore Shortcuts and Previous Outputs	15
Suppressing Output	15
Related Magic Commands	16
IPython and Shell Commands	16
Quick Introduction to the Shell	16
Shell Commands in IPython	18

Passing Values to and from the Shell	18
Shell-Related Magic Commands	19
Errors and Debugging	20
Controlling Exceptions: %xmode	20
Debugging: When Reading Tracebacks Is Not Enough	22
Profiling and Timing Code	25
Timing Code Snippets: %timeit and %time	25
Profiling Full Scripts: %prun	27
Line-by-Line Profiling with %lprun	28
Profiling Memory Use: %memit and %mprun	29
More IPython Resources	30
Web Resources	30
Books	31
2. Introduction to NumPy.....	33
Understanding Data Types in Python	34
A Python Integer Is More Than Just an Integer	35
A Python List Is More Than Just a List	37
Fixed-Type Arrays in Python	38
Creating Arrays from Python Lists	39
Creating Arrays from Scratch	39
NumPy Standard Data Types	41
The Basics of NumPy Arrays	42
NumPy Array Attributes	42
Array Indexing: Accessing Single Elements	43
Array Slicing: Accessing Subarrays	44
Reshaping of Arrays	47
Array Concatenation and Splitting	48
Computation on NumPy Arrays: Universal Functions	50
The Slowness of Loops	50
Introducing UFuncs	51
Exploring NumPy's UFuncs	52
Advanced Ufunc Features	56
Ufuncs: Learning More	58
Aggregations: Min, Max, and Everything in Between	58
Summing the Values in an Array	59
Minimum and Maximum	59
Example: What Is the Average Height of US Presidents?	61
Computation on Arrays: Broadcasting	63
Introducing Broadcasting	63
Rules of Broadcasting	65
Broadcasting in Practice	68

Comparisons, Masks, and Boolean Logic	70
Example: Counting Rainy Days	70
Comparison Operators as ufuncs	71
Working with Boolean Arrays	73
Boolean Arrays as Masks	75
Fancy Indexing	78
Exploring Fancy Indexing	79
Combined Indexing	80
Example: Selecting Random Points	81
Modifying Values with Fancy Indexing	82
Example: Binning Data	83
Sorting Arrays	85
Fast Sorting in NumPy: np.sort and np.argsort	86
Partial Sorts: Partitioning	88
Example: k-Nearest Neighbors	88
Structured Data: NumPy's Structured Arrays	92
Creating Structured Arrays	94
More Advanced Compound Types	95
RecordArrays: Structured Arrays with a Twist	96
On to Pandas	96
3. Data Manipulation with Pandas.....	97
Installing and Using Pandas	97
Introducing Pandas Objects	98
The Pandas Series Object	99
The Pandas DataFrame Object	102
The Pandas Index Object	105
Data Indexing and Selection	107
Data Selection in Series	107
Data Selection in DataFrame	110
Operating on Data in Pandas	115
Ufuncs: Index Preservation	115
UFuncs: Index Alignment	116
Ufuncs: Operations Between DataFrame and Series	118
Handling Missing Data	119
Trade-Offs in Missing Data Conventions	120
Missing Data in Pandas	120
Operating on Null Values	124
Hierarchical Indexing	128
A Multiply Indexed Series	128
Methods of MultiIndex Creation	131
Indexing and Slicing a MultiIndex	134

Rearranging Multi-Indices	137
Data Aggregations on Multi-Indices	140
Combining Datasets: Concat and Append	141
Recall: Concatenation of NumPy Arrays	142
Simple Concatenation with pd.concat	142
Combining Datasets: Merge and Join	146
Relational Algebra	146
Categories of Joins	147
Specification of the Merge Key	149
Specifying Set Arithmetic for Joins	152
Overlapping Column Names: The suffixes Keyword	153
Example: US States Data	154
Aggregation and Grouping	158
Planets Data	159
Simple Aggregation in Pandas	159
GroupBy: Split, Apply, Combine	161
Pivot Tables	170
Motivating Pivot Tables	170
Pivot Tables by Hand	171
Pivot Table Syntax	171
Example: Birthrate Data	174
Vectorized String Operations	178
Introducing Pandas String Operations	178
Tables of Pandas String Methods	180
Example: Recipe Database	184
Working with Time Series	188
Dates and Times in Python	188
Pandas Time Series: Indexing by Time	192
Pandas Time Series Data Structures	192
Frequencies and Offsets	195
Resampling, Shifting, and Windowing	196
Where to Learn More	202
Example: Visualizing Seattle Bicycle Counts	202
High-Performance Pandas: eval() and query()	208
Motivating query() and eval(): Compound Expressions	209
pandas.eval() for Efficient Operations	210
DataFrame.eval() for Column-Wise Operations	211
DataFrame.query() Method	213
Performance: When to Use These Functions	214
Further Resources	215

4. Visualization with Matplotlib.....	217
General Matplotlib Tips	218
Importing matplotlib	218
Setting Styles	218
show() or No show()? How to Display Your Plots	218
Saving Figures to File	221
Two Interfaces for the Price of One	222
Simple Line Plots	224
Adjusting the Plot: Line Colors and Styles	226
Adjusting the Plot: Axes Limits	228
Labeling Plots	230
Simple Scatter Plots	233
Scatter Plots with plt.plot	233
Scatter Plots with plt.scatter	235
plot Versus scatter: A Note on Efficiency	237
Visualizing Errors	237
Basic Errorbars	238
Continuous Errors	239
Density and Contour Plots	241
Visualizing a Three-Dimensional Function	241
Histograms, Binnings, and Density	245
Two-Dimensional Histograms and Binnings	247
Customizing Plot Legends	249
Choosing Elements for the Legend	251
Legend for Size of Points	252
Multiple Legends	254
Customizing Colorbars	255
Customizing Colorbars	256
Example: Handwritten Digits	261
Multiple Subplots	262
plt.axes: Subplots by Hand	263
plt.subplot: Simple Grids of Subplots	264
plt.subplots: The Whole Grid in One Go	265
plt.GridSpec: More Complicated Arrangements	266
Text and Annotation	268
Example: Effect of Holidays on US Births	269
Transforms and Text Position	270
Arrows and Annotation	272
Customizing Ticks	275
Major and Minor Ticks	276
Hiding Ticks or Labels	277
Reducing or Increasing the Number of Ticks	278

Fancy Tick Formats	279
Summary of Formatters and Locators	281
Customizing Matplotlib: Configurations and Stylesheets	282
Plot Customization by Hand	282
Changing the Defaults: rcParams	284
Stylesheets	285
Three-Dimensional Plotting in Matplotlib	290
Three-Dimensional Points and Lines	291
Three-Dimensional Contour Plots	292
Wireframes and Surface Plots	293
Surface Triangulations	295
Geographic Data with Basemap	298
Map Projections	300
Drawing a Map Background	304
Plotting Data on Maps	307
Example: California Cities	308
Example: Surface Temperature Data	309
Visualization with Seaborn	311
Seaborn Versus Matplotlib	312
Exploring Seaborn Plots	313
Example: Exploring Marathon Finishing Times	322
Further Resources	329
Matplotlib Resources	329
Other Python Graphics Libraries	330
5. Machine Learning	331
What Is Machine Learning?	332
Categories of Machine Learning	332
Qualitative Examples of Machine Learning Applications	333
Summary	342
Introducing Scikit-Learn	343
Data Representation in Scikit-Learn	343
Scikit-Learn’s Estimator API	346
Application: Exploring Handwritten Digits	354
Summary	359
Hyperparameters and Model Validation	359
Thinking About Model Validation	359
Selecting the Best Model	363
Learning Curves	370
Validation in Practice: Grid Search	373
Summary	375
Feature Engineering	375