Extracted from:

Python Companion to Data Science

Collect → Organize → Explore → Predict → Value

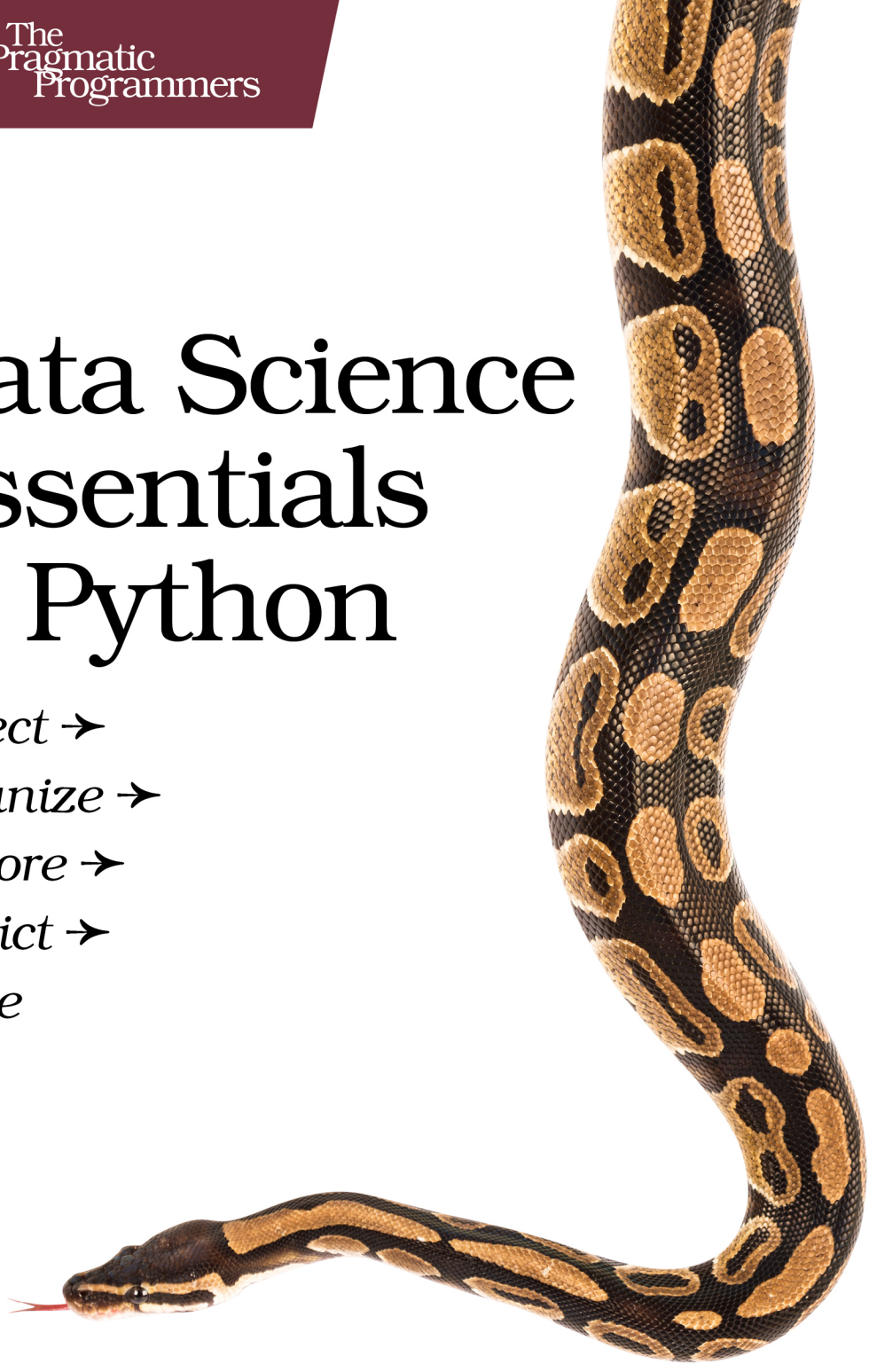
The Pragmatic Bookshelf

Raleigh, North Carolina

The Pragmatic Programmers

Data Science Essentials in Python

Collect ➔
Organize ➔
Explore ➔
Predict ➔
Value



Dmitry Zinoviev
edited by Katharine Dvorak

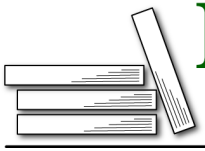
Python Companion to Data Science

Collect → Organize → Explore → Predict → Value

Dmitry Zinoviev

The Pragmatic Bookshelf

Raleigh, North Carolina



Pragmatic Bookshelf

Our Pragmatic books, screencasts, and audio books can help you and your team create better software and have more fun. Visit us at *https://pragprog.com*.

The team that produced this book includes:

Katharine Dvorak (editor)
Potomac Indexing, LLC (index)
Nicole Abramowitz (copyedit)
Gilson Graphics (layout)
Janet Furlow (producer)

For sales, volume licensing, and support, please contact *support@pragprog.com*.

For international rights, please contact *rights@pragprog.com*.

Printed in the United States of America.
ISBN-13: 978-1-68050-184-1
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—August 2016

To my beautiful and most intelligent wife Anna; to our children: graceful ballerina Eugenia and romantic gamer Roman; and to my first data science class of summer 2015.

I must instruct you in a little science by-and-by, to distract your thoughts.

➤ *Marie Corelli, British novelist*

Preface

This book was inspired by an introductory data science course in Python that I taught in summer 2015 to a small group of select undergraduate students of Suffolk University in Boston. The course was expected to be the first in a two-course sequence, with an emphasis on obtaining, cleaning, organizing, and visualizing data, sprinkled with some elements of statistics, machine learning, and network analysis.

I quickly came to realize that the abundance of systems and Python modules involved in these operations (databases, natural language processing frameworks, JSON and HTML parsers, and high-performance numerical data structures, to name a few) could easily overwhelm not only an undergraduate student, but also a seasoned professional. In fact, I have to confess that while working on my own research projects in the fields of data science and network analysis, I had to spend more time calling the `help()` function and browsing scores of online Python discussion boards than I was comfortable with. In addition, I must admit to some embarrassing moments in the classroom when I seemed to have hopelessly forgotten the name of some function or some optional parameter.

As a part of teaching the course, I compiled a set of cheat sheets on various topics that turned out to be a useful reference. The cheat sheets eventually evolved into this book. Hopefully, having it on your desk will make you think more about data science and data analysis than about function names and optional parameters.

About This Book

This book covers data acquisition, cleaning, storing, retrieval, transformation, visualization, elements of advanced data analysis (network analysis), statistics, and machine learning. It is not an introduction to data science or a general data science reference, although you'll find a quick overview of how to do data science in Chapter 1, *What Is Data Science?*, on page ?. I assume that you

have learned the methods of data science, including statistics, elsewhere. The subject index at the end of the book refers to the Python implementations of the key concepts, but in most cases you will already be familiar with the concepts.

You'll find a summary of Python data structures; string, file, and web functions; regular expressions; and even list comprehension in Chapter 2, *Core Python for Data Science*, on page ?. This summary is provided to refresh your knowledge of these topics, not to teach them. There are a lot of excellent Python texts, and having a mastery of the language is absolutely important for a successful data scientist.

The first part of the book looks at working with different types of text data, including processing structured and unstructured text, processing numeric data with the NumPy and Pandas modules, and network analysis. Three more chapters address different analysis aspects: working with relational and non-relational databases, data visualization, and simple predictive analysis.

This book is partly a story and partly a reference. Depending on how you see it, you can either read it sequentially or jump right to the index, find the function or concept of concern, and look up relevant explanations and examples. In the former case, if you are an experienced Python programmer, you can safely skip Chapter 2, *Core Python for Data Science*, on page ?. If you do not plan to work with external databases (such as `MySQL`), you can ignore Chapter 4, *Working with Databases*, on page ?, as well. Lastly, Chapter 9, *Probability and Statistics*, on page ?, assumes that you have no idea about statistics. If you do, you have an excuse to bypass the first two units and find yourself at Unit 47, *Doing Stats the Python Way*, on page ?.

About the Audience

At this point, you may be asking yourself if you want to have this book on your bookshelf.

The book is intended for graduate and undergraduate students, data science instructors, entry-level data science professionals—especially those converting from R to Python—and developers who want a reference to help them remember all of the Python functions and options.

Is that you? If so, *abandon all hesitation and enter.*

About the Software

Despite some controversy surrounding the transition from Python 2.7 to Python 3.3 and above, I firmly stand behind the newer Python dialect. Most new Python software is developed for 3.3, and most of the legacy software has been successfully ported to 3.3, too. Considering the trend, it would be unwise to choose an outdated dialect, no matter how popular it may seem at the time.

All Python examples in this book are known to work for the modules mentioned in the following table. All of these modules, with the exception of the community module that must be installed separately[1] and the Python interpreter itself, are included in the Anaconda distribution, which is provided by Continuum Analytics and is available for free.[2]

Package	Used version	Package	Used version
BeautifulSoup	4.3.2	community	0.3
json	2.0.9	html5lib	0.999
matplotlib	1.4.3	networkx	1.10.0
nltk	3.1.0	numpy	1.10.1
pandas	0.17.0	pymongo	3.0.2
pymysql	0.6.2	python	3.4.3
scikit-learn	0.16.1	scipy	0.16.0

Table 1—Software Components Used in the Book

If you plan to experiment (or actually work) with databases, you will also need to download and install MySQL[3] and MongoDB.[4] Both databases are free and known to work on Linux, Mac OS, and Windows platforms.

Notes on Quotes

Python allows the user to enclose character strings in 'single', "double", '''triple''', and even """triple double""" quotes (the latter two can be used for multiline strings). However, when printing out strings, it always uses single quote notation, regardless of which quotes you used in the program.

Many other languages (C, C++, Java) use single and double quotes differently: single for individual characters, double for character strings. To pay tribute

1. pypi.python.org/pypi/python-louvain/0.3
2. www.continuum.io
3. www.mysql.com
4. www.mongodb.com

to this differentiation, in this book I, too, use single quotes for single characters and double quotes for character strings.

The Book Forum

The community forum for this book can be found online at the Pragmatic Programmers web page for this book.[5] There you can ask questions, post comments, and submit errata.

Another great resource for questions and answers (not specific to this book) is the newly created Data Science Stack Exchange forum.[6]

Your Turn

The end of each chapter features a unit called "Your Turn." This unit has descriptions of several projects that you may want to accomplish on your own (or with someone you trust) to strengthen your understanding of the material.

The projects marked with a single star* are the simplest. All you need to work on them is solid knowledge of the functions mentioned in the preceding chapters. Expect to complete single-star projects in no more than thirty minutes. You'll find solutions to them in Appendix 2, *Solutions to Single-Star Projects*, on page ?.

The projects marked with two stars** are hard(er). They may take you an hour or more, depending on your programming skills and habits. Two-star projects involve the use of intermediate data structures and well thought-out algorithms.

Finally, the three-star*** projects are the hardest. Some of the three-star projects may not even have a perfect solution, so don't get desperate if you cannot find one! Just by working on these projects, you certainly make yourself a better programmer and a better data scientist. And if you're an educator, think of the three-star projects as potential mid-semester assignments.

Now, let's get started!

Dmitry Zinoviev
dzinoviev@gmail.com
August 2016

5. pragprog.com/book/dzpyds
6. datascience.stackexchange.com