Quick answers to common problems

Python Data Visualization Cookbook

Over 60 recipes that will enable you to learn how to create attractive visualizations using Python's most popular libraries

Igor Milovanović

[PACKT] PUBLISHING open source community experience distilled

Python Data Visualization Cookbook

Over 60 recipes that will enable you to learn how to create attractive visualizations using Python's most popular libraries

Igor Milovanović

[PACKT] open source
community experience distilled
PUBLISHING

BIRMINGHAM - MUMBAI

Python Data Visualization Cookbook

Copyright © 2013 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: November 2013

Production Reference: 1191113

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-78216-336-7

www.packtpub.com

Cover Image by Gorkee Bhardwaj (afterglowpictures@gmail.com)

Credits

Author

Igor Milovanović

Reviewers

Tarek Amr

Simeone Franklin

Jayesh K. Gupta

Kostiantyn Kucher

Kenneth Emeka Odoh

Acquisition Editor

James Jones

Lead Technical Editor

Ankita Shashi

Technical Editors

Pratik More

Amit Ramadas

Ritika Singh

Copy Editors

Brandt D'Mello

Janbal Dharmaraj

Deepa Nambiar

Kirti Pai

Laxmi Subramanian

Project Coordinator

Rahul Dixit

Proofreaders

Amy Johnson

Lindsey Thomas

Indexer

Mariammal Chettiyar

Graphics

Abhinash Sahu

Production Coordinator

Shantanu Zagade

Cover Work

Shantanu Zagade

About the Author

Igor Milovanović is an experienced developer with a strong background in Linux system and software engineering. He has skills in building scalable data-driven distributed software-rich systems.

He is an Evangelist for high-quality systems design who holds strong interests in software architecture and development methodologies. He is always persistent on advocating methodologies that promote high-quality software, such as test-driven development, one-step builds, and continuous integration.

He also possesses solid knowledge of product development. Having field experience and official training, he is capable of transferring knowledge and communication flow from business to developers and vice versa.

I am most grateful to my fiance for letting me spend endless hours on the work instead with her and for being an avid listener to my endless book monologues. I want to also thank my brother for always being my strongest supporter. I am thankful to my parents for letting me develop myself in various ways and become the person I am today.

I could not write this book without enormous energy from open source community that developed Python, matplotlib, and all libraries that we have used in this book. I owe the most to the people behind all these projects. Thank you.

About the Reviewers

Tarek Amr achieved his postgraduate degree in Data Mining and Information Retrieval from the University of East Anglia. He has about 10 years' experience in Software Development. He has been volunteering in Global Voices Online (GVO) since 2007, and currently he is the local ambassador of the Open Knowledge Foundation (OKFN) in Egypt. Words such as Open Data, Government 2.0, Data Visualisation, Data Journalism, Machine Learning, and Natural Language Processing are like music to his ears.

Tarek's Twitter handle is `@gr33ndata` and his homepage is `http://tarekamr.appspot.com/`.

Jayesh K. Gupta is the Lead Developer of Matlab Toolbox for Biclustering Analysis (MTBA). He is currently an undergraduate student and researcher at IIT Kanpur. His interests lie in the field of pattern recognition. His interests also lie in basic sciences, recognizing them as the means of analyzing patterns in nature. Coming to IIT, he realized how this analysis is being augmented by Machine Learning algorithms with various diverse applications. He believes that augmenting human thought with machine intelligence is one of the best ways to advance human knowledge. He is a long time technophile and a free-software Evangelist. He usually goes by the handle, `rejuvyesh` online. He is also an avid reader and his books can be checked out at Goodreads. Checkout his projects at Bitbucket and GitHub. For all links visit `http://home.iitk.ac.in/~jayeshkg/`. He can be contacted at `a2z.jayesh@gmail.com`.

Kostiantyn Kucher was born in Odessa, Ukraine. He received his Master's degree in Computer Science from Odessa National Polytechnic University in 2012. He used Python as well as Matplotlib and PIL for Machine Learning and Image Recognition purposes.

Currently, Kostiantyn is a PhD student in Computer Science specializing in Information Visualization. He conducts his research under the supervision of *Prof. Dr. Andreas Kerren* with the ISOVIS group at the Computer Science Department of Linnaeus University (Växjö, Sweden).

Kenneth Emeka Odoh performs research on state of the art Data Visualization techniques. His research interest includes exploratory search where the users are guided to their search results using visual clues.

Kenneth is proficient in Python programming. He has presented a Python conference talk at Pycon, Finland in 2012 where he spoke about Data Visualization in Django to a packed audience.

He currently works as a Graduate Researcher at the University of Regina, Canada. He is a polyglot with experience in developing applications in C, C++, Python, and Java programming languages.

When Kenneth is not writing source codes, you can find him singing at the Campion College chant choir.

www.PacktPub.com

Support files, eBooks, discount offers and more

You might want to visit `www.PacktPub.com` for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at `www.PacktPub.com` and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at `service@packtpub.com` for more details.

At `www.PacktPub.com`, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.

`http://PacktLib.PacktPub.com`

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

Why Subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

Free Access for Packt account holders

If you have an account with Packt at `www.PacktPub.com`, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

Table of Contents

Preface

The best data is the data that we can see and understand. As developers, we want to create and build the most comprehensive and understandable visualizations. It is not always simple; we need to find the data, read it, clean it, massage it, and then use the right tool to visualize it. This book explains the process of how to read, clean, and visualize the data into information with straight and simple (and not so simple) recipes.

How to read local data, remote data, CSV, JSON, and data from relational databases are all explained in this book.

Some simple plots can be plotted with a simple one-liner in Python using matplotlib, but doing more advanced charting requires knowledge of more than just Python. We need to understand the information theory and human perception aesthetics to produce the most appealing visualizations.

This book will explain some practices behind plotting with matplotlib in Python, statistics used, and usage examples for different charting features we should use in an optimal way.

This book is written and the code is developed on Ubuntu 12.03 using Python 2.7, IPython 0.13.2, virtualenv 1.9.1, matplotlib 1.2.1, NumPy 1.7.1, and SciPy 0.11.0.

What this book covers

Chapter 1, *Preparing Your Working Environment*, covers a set of installation recipes and advices on how to install the required Python packages and libraries on your platform.

Chapter 2, *Knowing Your Data*, introduces you to common data formats and how to read and write them, be it CSV, JSON, XSL, or relational databases.

Chapter 3, *Drawing Your First Plots and Customizing Them*, starts with drawing simple plots and covers some of the customization.

Chapter 4, *More Plots and Customizations*, follows up from previous chapter and covers more advanced charts and grid customization.

Chapter 5, *Making 3D Visualizations*, covers three-dimensional data visualizations such as 3D bars, 3D histograms, and also matplotlib animations.

Chapter 6, *Plotting Charts with Images and Maps*, covers image processing, projecting data onto maps, and creating CAPTCHA test images.

Chapter 7, *Using Right Plots to Understand Data*, covers explanations and recipes on some more advanced plotting techniques such as spectrograms and correlations.

Chapter 8, *More on matplotlib Gems*, covers a set of charts such as Gantt charts, box plots, and whisker plots, and also explains how to use LaTeX for rendering text in matplotlib.

What you need for this book

For this book, you will need Python 2.7.3 or a later version installed on your operating system. This book was written using Ubuntu 12.03's Python default version (2.7.3).

Other software packages used in this book are IPython, which is an interactive Python environment that is very powerful, and flexible. This can be installed using package managers for Linux-based OSes or prepared installers for Windows and Mac OSes.

If you are new to Python installation and software installation in general, it is very much recommended to use prepackaged scientific Python distributions such as Anaconda, Enthought Python Distribution, or Python(X,Y).

Other required software mainly comprises of Python packages that are all installed using the Python installation manager, `pip`, which itself is installed using Python's `easy_install` setup tool.

Who this book is for

Python Data Visualization Cookbook is for developers who already know about Python programming in general. If you have heard about data visualization but don't know where to start, this book will guide you from the start and help you understand data, data formats, data visualization, and how to use Python to visualize data.

You will need to know some general programming concepts, and any kind of programming experience will be helpful. However, the code in this book is explained almost line by line. You don't need math for this book; every concept that is introduced is thoroughly explained in plain English, and references are available for further interest in the topic.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "We packed our little demo in class `DemoPIL`, so that we can extend it easily, while sharing the common code around the demo function, `run_fixed_filters_demo`."

A block of code is set as follows:

```
def _load_image(self, imfile):
    self.im = mplimage.imread(imfile)
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
# tidy up tick labels size
all_axes = plt.gcf().axes
for ax in all_axes:
    for ticklabel in ax.get_xticklabels() + ax.get_yticklabels():
        ticklabel.set_fontsize(10)
```

Any command-line input or output is written as follows:

```
$ sudo python setup.py install
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "We then set up a label for the stem plot and the position of baseline, which defaults to **0**."

Warnings or important notes appear in a box like this.

Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to `feedback@packtpub.com`, and mention the book title via the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on `www.packtpub.com/authors`.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at `http://www.packtpub.com`. If you purchased this book elsewhere, you can visit `http://www.packtpub.com/support` and register to have the files e-mailed directly to you.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting `http://www.packtpub.com/submit-errata`, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from `http://www.packtpub.com/support`.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at `copyright@packtpub.com` with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at `questions@packtpub.com` if you are having a problem with any aspect of the book, and we will do our best to address it.

1
Preparing Your Working Environment

In this chapter, we will cover the following recipes:

- Installing matplotlib, NumPy, and SciPy
- Installing virtualenv and virtualenvwrapper
- Installing matplotlib on Mac OS X
- Installing matplotlib on Windows
- Installing Python Imaging Library (PIL) for image processing
- Installing a requests module
- Customizing matplotlib's parameters in code
- Customizing matplotlib's parameters per project

Introduction

This chapter introduces the reader to the essential tooling and installation and configuration of them. This is a necessary work and common base for the rest of the book. If you have never used Python for data and image processing and visualization, it is advised not to skip this chapter. Even if you do skip it, you can always return to this chapter in case you need to install some supporting tool or verify what version you need to support the current solution.

Installing matplotlib, NumPy, and SciPy

This chapter describes several ways of installing **matplotlib** and required dependencies under Linux.

Getting ready

We assume that you already have Linux (preferably Debian/Ubuntu or RedHat/SciLinux) installed and Python installed on it. Usually, Python is already installed on the mentioned Linux distributions and, if not, it is easily installable through standard means. We assume that Python 2.7+ Version is installed on your workstation.

Almost all code should work with Python 3.3+ Versions, but because most operating systems still deliver Python 2.7 (some even Python 2.6) we decided to write the Python 2.7 Version code. The differences are small, mainly in version of packages and some code (xrange should be substituted with range in Python 3.3+).

We also assume that you know how to use your OS package manager in order to install software packages and know how to use a terminal.

Build requirements must be satisfied before matplotlib can be built.

matplotlib requires **NumPy**, **libpng**, and **freetype** as build dependencies. In order to be able to build matplotlib from source, we must have installed NumPy. Here's how to do it:

Install NumPy (at least 1.4+, or 1.5+ if you want to use it with Python 3) from `http://www.numpy.org/`.

NumPy will provide us with data structures and mathematical functions for using it with large datasets. Python's default data structures such as tuples, lists, or dictionaries are great for insertions, deletions, and concatenation. NumPy's data structures support "vectorized" operations and are very efficient for use and for executions. They are implemented with Big Data in mind and rely on C implementations that allow efficient execution time.

SciPy, building on top of NumPy, is the de facto standard's scientific and numeric toolkit for Python comprising great selection of special functions and algorithms, most of them actually implemented in C and Fortran, coming from the well-known Netlib repository (see `http://www.netlib.org`).

Perform the following steps for installing NumPy:

1. Install Python-NumPy package:

   ```
   $ sudo apt-get install python-numpy
   ```

2. Check the installed version:

   ```
   $ python -c 'import numpy; print numpy.__version__'
   ```

3. Install the required libraries:

   - **libpng 1.2**: PNG files support (requires zlib)
   - **freetype 1.4+**: True type font support

     ```
     $ sudo apt-get install build-dep python-matplotlib
     ```

   If you are using RedHat or variation of this distribution (Fedora, SciLinux, or CentOS) you can use yum to perform same installation:

   ```
   $ su -c 'yum-builddep python-matplotlib'
   ```

How to do it...

There are many ways one can install matplotlib and its dependencies: from source, from precompiled binaries, from OS package manager, and with prepackaged python distributions with built-in matplotlib.

Most probably the easiest way is to use your distribution's package manager. For Ubuntu that should be:

```
# in your terminal, type:
$ sudo apt-get install python-numpy python-matplotlib python-scipy
```

If you want to be on the bleeding edge, the best option is to install from source. This path comprises a few steps: Get the source, build requirements, and configure, compile, and install.

Download the latest source from code host `www.github.com` by following these steps:

```
$ cd ~/Downloads/
$ wget https://github.com/downloads/matplotlib/matplotlib/matplotlib-
1.2.0.tar.gz
$ tar xzf matplotlib-1.2.0.tar.gz
$ cd matplotlib-1.2.0
$ python setup.py build
$ sudo python setup.py install
```

Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at `http://www.packtpub.com`. If you purchased this book elsewhere, you can visit `http://www.packtpub.com/support` and register to have the files e-mailed directly to you.

How it works...

We use standard **Python Distribution Utilities**, known as **Distutils**, to install matplotlib from source code. This procedure requires us to previously install dependencies, as we already explained in the *Getting ready* section of this recipe. The dependencies are installed using the standard Linux packaging tools.

There's more...

There are more optional packages that you might want to install depending on what your data visualization projects are about.

No matter what project you are working on, we recommend installing IPython—an Interactive Python shell that supports PyLab mode where you already have matplotlib and related packages, such as NumPy and SciPy, imported and ready to play with! Please refer to IPython's official site on how to install it and use it—it is, though, very straightforward.

Installing virtualenv and virtualenvwrapper

If you are working on many projects simultaneously, or even just switching between them frequently, you'll find that having everything installed system-wide is not the best option and can bring problems in future on different systems (production) where you want to run your software. This is not a good time to find out that you are missing a certain package or have versioning conflicts between packages that are already installed on production system; hence, virtualenv.

virtualenv is an open source project started by *Ian Bicking* that enables a developer to isolate working environments per project, for easier maintenance of different package versions.

For example, you inherited legacy Django website based on Django 1.1 and Python 2.3, but at the same time you are working on a new project that must be written in Python 2.6. This is my usual case—having more than one required Python version (and related packages) depending on the project I am working on.

virtualenv enables me to easily switch to different environments and have the same package easily reproduced if I need to switch to another machine or to deploy software to a production server (or to a client's workstation).

Getting ready

To install virtualenv, you must have workable installation of Python and pip. Pip is a tool for installing and managing Python packages, and it is a replacement for easy install. We will use pip through most of this book for package management. Pip is easily installed, as root executes the following line in your terminal:

```
# easy_install pip
```

virtualenv by itself is really useful, but with the help of virtualenvwrapper, all this becomes easy to do and also easy to organize many virtual environments. See all the features at `http://virtualenvwrapper.readthedocs.org/en/latest/#features`.

How to do it...

By performing the following steps you can install the virtualenv and virtualenvwrapper tools:

1. Install virtualenv and virtualenvwrapper:

```
$ sudo pip virtualenv
$ sudo pip virtualenvwrapper
# Create folder to hold all our virtual environments and export
the path to it.
$ export VIRTENV=~/.virtualenvs
$ mkdir -p $VIRTENV
# We source (ie. execute) shell script to activate the wrappers
$ source /usr/local/bin/virtualenvwrapper.sh
# And create our first virtual environment
$ mkvirtualenv virt1
```

2. You can now install our favorite package inside `virt1`:

```
(virt1)user1:~$ pip install matplotlib
```

3. You will probably want to add the following line to your `~/.bashrc` file:

```
source /usr/loca/bin/virtualenvwrapper.sh
```

Few useful and most frequently used commands are as follows:

- `mkvirtualenv ENV`: This creates virtual environment with name ENV and activates it
- `workon ENV`: This activates the previously created ENV
- `deactivate`: This gets us out of the current virtual environment

Installing matplotlib on Mac OS X

The easiest way to get matplotlib on Mac OS X is to use prepackaged python distributions such as **Enthought Python Distribution** (**EPD**). Just go to the EPD site and download and install the latest stable version for your OS.

In case you are not satisfied with EPD or cannot use it for other reasons such as versions distributed with it, there is a manual (read: harder) way of installing Python, matplotlib, and its dependencies.

Getting ready

We will use the **Homebrew** project that eases installation of all software that Apple did not install on your OS, including Python and matplotlib. Under the hood, Homebrew is a set of Ruby and Git that automate download and installation. Following these instructions should get the installation working. First, we will install Homebrew, and then Python, followed by tools such as virtualenv, then dependencies for matplotlib (NumPy and SciPy), and finally matplotlib. Hold on, here we go.

How to do it...

1. In your Terminal paste and execute the following command:

   ```
   ruby <(curl -fsSkL raw.github.com/mxcl/homebrew/go)
   ```

   After the command finishes, try running brew update or brew doctor to verify that installation is working properly.

2. Next, add the Homebrew directory to your system path, so the packages you install using Homebrew have greater priority than other versions. Open `~/.bash_profile` (or `/Users/[your-user-name]/.bash_profile`) and add the following line to the end of file:

   ```
   export PATH=/usr/local/bin:$PATH
   ```

3. You will need to restart the terminal so it picks a new path. Installing Python is as easy as firing up another one-liner:

   ```
   brew install python --framework --universal
   ```

   This will also install any prerequisites required by Python.

4. Now, you need to update your path (add to the same line):

   ```
   export PATH=/usr/local/share/python:/usr/local/bin:$PATH
   ```

5. To verify that installation worked, type `python --version` at the command line, you should see `2.7.3` as the version number in the response.

6. You should have pip installed by now. In case it is not installed, use `easy_install` to add pip:

```
$ easy_install pip
```

7. Now, it's easy to install any required package; for example, virtualenv and virtualenvwrapper are useful:

```
pip install virtualenv
pip install virtualenvwrapper
```

8. Next step is what we really wanted to do all along—install matplotlib:

```
pip install numpy
brew install gfortran
pip install scipy
```

Mountain Lion users will need to install the development version of SciPy (0.11) by executing the following line:

```
pip install -e git+https://github.com/scipy/scipy#egg=scipy-dev
```

9. Verify that everything is working. Call Python and execute the following commands:

```
import numpy
print numpy.__version__
import scipy
print scipy.__version__
quit()
```

10. Install matplotlib:

```
pip install matplotlib
```

Installing matplotlib on Windows

In this recipe, we will demonstrate how to install Python and start working with matplotlib installation. We assume Python was not previously installed.

Getting ready

There are two ways of installing matplotlib on Windows. The easier way is by installing prepackaged Python environments such as EPD, Anaconda and Python(x,y). This is the suggested way to install Python, especially for beginners.

sample content of Python Data Visualization Cookbook - Second Edition

- The Kaiser and His Court: Wilhelm II and the Government of Germany pdf
- *Handbook of Exchange Rates (Wiley Handbooks in Financial Engineering and Econometrics, Volume 2) pdf*
- *read Accepting Authoritarianism: State-Society Relations in China's Reform Era*
- Das Licht von Terrania (Perry Rhodan Neo, Band 85; Kampfzone Erde, Band 1) pdf, azw (kindle)

- http://rodrigocaporal.com/library/Silencing-Eve--Eve-Duncan--Book-18-.pdf
- http://thermco.pl/library/Things-I-Shouldn-t-Think.pdf
- http://www.celebritychat.in/?ebooks/All-Is-Well--Heal-Your-Body-with-Medicine--Affirmations--and-Intuition.pdf
- http://flog.co.id/library/The-Tears-of-Re--Beekeeping-in-Ancient-Egypt.pdf