

# Python Data Mining Quick Start Guide

A beginner's guide to extracting valuable insights from your data



Nathan Greeneltch

**Packt**>

[www.packt.com](http://www.packt.com)

# Python Data Mining Quick Start Guide

A beginner's guide to extracting valuable insights  
from your data

**Nathan Greeneltch**

**Packt**>

**BIRMINGHAM - MUMBAI**

# Python Data Mining Quick Start Guide

Copyright © 2019 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 or its dealers and distributors, will be held liable for any damages caused or alleged to have been 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.

**Commissioning Editor:** Amey Varangaonkar

**Acquisition Editor:** Reshma Raman

**Content Development Editor:** Smit Carvalho

**Technical Editor:** Diksha Wakode

**Copy Editor:** Safis Editing

**Project Coordinator:** Kinjal Bari

**Proofreader:** Safis Editing

**Indexer:** Pratik Shirodkar

**Graphics:** Alishon Mendonsa

**Production Coordinator:** Jayalaxmi Raja

First published: April 2019

Production reference: 1240419

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham

B3 2PB, UK.

ISBN 978-1-78980-026-5

[www.packtpub.com](http://www.packtpub.com)

*To my children, Vaughn and Vera. They are the lit candle around every corner I turn.*



mapt.io

Mapt is an online digital library that gives you full access to over 5,000 books and videos, as well as industry leading tools to help you plan your personal development and advance your career. For more information, please visit our website.

## Why subscribe?

- Spend less time learning and more time coding with practical eBooks and Videos from over 4,000 industry professionals
- Improve your learning with Skill Plans built especially for you
- Get a free eBook or video every month
- Mapt is fully searchable
- Copy and paste, print, and bookmark content

## Packt.com

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.packt.com](http://www.packt.com) and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at [customercare@packtpub.com](mailto:customercare@packtpub.com) for more details.

At [www.packt.com](http://www.packt.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.

# Contributors

## About the author

**Nathan Greeneltch, PhD** is a ML engineer at Intel Corp and resident data mining and analytics expert in the AI consulting group. He's worked with Python analytics in both the start-up realm and the large-scale manufacturing sector over the course of the last decade. Nathan regularly mentors new hires and engineers fresh to the field of analytics, with impromptu chalk talks and division-wide knowledge-sharing sessions at Intel. In his past life, he was a physical chemist studying surface enhancement of the vibration signals of small molecules; a topic on which he wrote a doctoral thesis while at Northwestern University in Evanston, IL. Nathan hails from the southeastern United States, with family in equal parts from Arkansas and Florida.

*I'd like to thank my loving wife, Lei-Ann, who encouraged me from conception to completion of this book. It would not have been written without her. Also, I'd like to thank the students and colleagues that contributed to the narrative style and teaching methods I've adopted over the years. The list is long, but to name a few: Martin, Michael, Liu, Nan, Preethi, and Britt. Lastly, thank you to my family for supporting me and injecting the perspective that I share in this book.*

## About the reviewer

**Julian Quick** is a graduate student attending the University of Colorado Boulder, studying turbulent flow modeling. Julian is interested in the optimization and uncertainty quantification of wind energy systems.

*I would like to thank my advisors, Peter Hamlington and Ryan King, for their continued patience and encouragement.*

## Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit [authors.packtpub.com](https://authors.packtpub.com) and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

# Table of Contents

<b>Preface</b>	1
<b>Chapter 1: Data Mining and Getting Started with Python Tools</b>	6
<b>Descriptive, predictive, and prescriptive analytics</b>	8
<b>What will and will not be covered in this book</b>	9
Recommended readings for further explanation	9
<b>Setting up Python environments for data mining</b>	10
<b>Installing the Anaconda distribution and Conda package manager</b>	11
Installing on Linux	11
Installing on Windows	12
Installing on macOS	12
<b>Launching the Spyder IDE</b>	12
<b>Launching a Jupyter Notebook</b>	14
<b>Installing high-performance Python distribution</b>	16
<b>Recommended libraries and how to install</b>	17
Recommended libraries	18
<b>Summary</b>	19
<b>Chapter 2: Basic Terminology and Our End-to-End Example</b>	20
<b>Basic data terminology</b>	20
Sample spaces	21
Variable types	21
Data types	22
<b>Basic summary statistics</b>	23
<b>An end-to-end example of data mining in Python</b>	24
Loading data into memory – viewing and managing with ease using pandas	24
Plotting and exploring data – harnessing the power of Seaborn	25
Transforming data – PCA and LDA with scikit-learn	29
Quantifying separations – k-means clustering and the silhouette score	34
Making decisions or predictions	35
<b>Summary</b>	38
<b>Chapter 3: Collecting, Exploring, and Visualizing Data</b>	39
<b>Types of data sources and loading into pandas</b>	39
Databases	40
Basic Structured Query Language (SQL) queries	41
Disks	44
Web sources	44
From URLs	45
From Scikit-learn and Seaborn-included sets	45



<b>Access, search, and sanity checks with pandas</b>	45
<b>Basic plotting in Seaborn</b>	51
<b>Popular types of plots for visualizing data</b>	51
Scatter plots	52
Histograms	54
Jointplots	56
Violin plots	57
Pairplots	59
<b>Summary</b>	61
<b>Chapter 4: Cleaning and Readying Data for Analysis</b>	62
<b>The scikit-learn transformer API</b>	62
<b>Cleaning input data</b>	64
Missing values	64
Finding and removing missing values	65
Imputing to replace the missing values	67
Feature scaling	68
Normalization	69
Standardization	70
Handling categorical data	71
Ordinal encoding	71
One-hot encoding	73
Label encoding	75
<b>High-dimensional data</b>	76
<b>Dimension reduction</b>	76
Feature selection	76
Feature filtering	77
The variance threshold	77
The correlation coefficient	79
Wrapper methods	81
Sequential feature selection	81
Transformation	83
PCA	84
LDA	86
<b>Summary</b>	88
<b>Chapter 5: Grouping and Clustering Data</b>	89
<b>Introducing clustering concepts</b>	90
Location of the group	92
Euclidean space (centroids)	92
Non-Euclidean space (medioids)	93
Similarity	93
Euclidean space	94
The Euclidean distance	94
The Manhattan distance	94
Maximum distance	95
Non-Euclidean space	95
The cosine distance	96

The Jaccard distance	96
Termination condition	96
With known number of groupings	97
Without known number of groupings	97
Quality score and silhouette score	97
<b>Clustering methods</b>	98
Means separation	101
K-means	102
Finding k	103
K-means++	104
Mini batch K-means	105
Hierarchical clustering	105
Reuse the dendrogram to find number of clusters	109
Plot dendrogram	110
Density clustering	110
Spectral clustering	112
<b>Summary</b>	115
<b>Chapter 6: Prediction with Regression and Classification</b>	116
<b>Scikit-learn Estimator API</b>	116
<b>Introducing prediction concepts</b>	117
Prediction nomenclature	119
Mathematical machinery	120
Loss function	120
Gradient descent	122
Fit quality regimes	126
<b>Regression</b>	127
Metrics of regression model prediction	127
Regression example dataset	128
Linear regression	129
Extension to multivariate form	130
Regularization with penalized regression	131
Regularization penalties	132
<b>Classification</b>	134
Classification example dataset	135
Metrics of classification model prediction	136
Multi-class classification	136
One-versus-all	137
One-versus-one	137
Logistic regression	138
Regularized logistic regression	141
Support vector machines	141
Soft-margin with C	145
The kernel trick	145
Tree-based classification	146
Decision trees	147
Node splitting with Gini	148
Random forest	149

*Table of Contents*

---

Avoid overfitting and speed up the fits	150
Built-in validation with bagging	150
<b>Tuning a prediction model</b>	152
Cross-validation	152
Introduction of the validation set	153
Multiple validation sets with k-fold method	154
Grid search for hyperparameter tuning	155
<b>Summary</b>	156
<b>Chapter 7: Advanced Topics - Building a Data Processing Pipeline and Deploying It</b>	157
<b>Pipelining your analysis</b>	157
Scikit-learn's pipeline object	158
<b>Deploying the model</b>	160
Serializing a model and storing with the pickle module	161
Loading a serialized model and predicting	161
<b>Python-specific deployment concerns</b>	161
<b>Summary</b>	162
<b>Other Books You May Enjoy</b>	163
<b>Index</b>	166

---

# Preface

This book introduces data mining with popular free Python libraries. It is written in a conversational style, aiming to be approachable while imparting intuition on the reader. Data mining is a broad field of analytical methods designed to uncover insights from your data that are not obvious or discoverable by conventional analysis techniques. The field of data mining is vast, so the topics in this quick start guide were chosen by their relevance to not only their field of origin, but also the adjacent applications of machine learning and artificial intelligence. After a procedural first half, focused on getting the reader comfortable with data collection, loading, and munging, the book will move to a completely conceptual discussion. The concepts are introduced from first principles intuition and broadly grouped as transformation, clustering, and prediction. Popular methods such as principal component analysis, k-means clustering, support vector machines, and random forest are all covered in the conceptual second half of the book. The book ends with a discussion on pipe-lining and deploying your analytical models.

## Who this book is for

This book is targeted at individuals who are new to the field of data mining and analytics with Python. Very little background is assumed in Python programming or math above the high-school level. All of the Python libraries used in the book are freely available at no cost on a variety of platforms, so anyone with access to the internet should be able to learn and practice the concepts introduced.

## What this book covers

The first three and a half chapters of the book are focused on the procedural nuts and bolts of a data mining project. This includes creating a data mining Python environment, loading data from a variety of sources, and munging the data for downstream analysis. The remaining content in the book is mostly conceptual, and delivered in a conversational style very close to how I would train a new hire at my company.

Chapter 1, *Data Mining and Getting Started with Python Tools*, covers the topic of getting started with your software environment. It also covers how to download and install high-speed Python and popular libraries such as `pandas`, `scikit-learn`, and `seaborn`. After reading this chapter and setting up your environment, you should be ready to follow along with the demonstrations throughout the rest of the book.

Chapter 2, *Basic Terminology and our End-to-End Example*, covers the basic statistics and data terminology that are required for working in data mining. The final portion of the chapter is dedicated to a full working example, which combined the types of techniques that will be introduced later on in this book. You will also have a better understanding of the thought processes behind analysis and the common steps taken to address a problem statement that you may encounter in the field.

Chapter 3, *Collecting, Exploring, and Visualizing Data*, covers the basics of loading data from databases, disks, and web sources. It also covers the basic SQL queries, and pandas' access and search functions. The last sections of the chapter introduce the common types of plots using Seaborn.

Chapter 4, *Cleaning and Readying Data for Analysis*, covers the basics of data cleanup and dimensionality reduction. After reading it, you will understand how to work with missing values, rescale input data, and handle categorical variables. You will also understand the troubles of high-dimensional data, and how to combat this with feature reduction techniques including filter, wrapper, and transformation methods.

Chapter 5, *Grouping and Clustering Data*, introduces the background and thought processes that goes into designing a clustering algorithm for data mining work. It then introduces common clustering methods in the field and carries out a comparison between all of them with toy datasets. After reading this chapter, you will know the difference between algorithms that cluster based on means separation, density, and connectivity. You will also be able to look at a plot of incoming data and have some intuition on whether clustering will fit your mining project.

Chapter 6, *Prediction with Regression and Classification*, covers the basics behind using a computer to learn prediction models by introducing the loss function and gradient descent. It then introduces the concepts of overfitting, underfitting, and the penalty approach to regularize your model during fits. It also covers common regression and classification techniques, and the regularized versions of each of these where appropriate. The chapter finishes with a discussion of best practices for model tuning, including cross-validation and grid search.

Chapter 7, *Advanced Topics – Building a Data Processing Pipeline and Deploying*, This chapter covers a strategy for pipe-lining and deploying using built-in Scikit-learn methods. It also introduces the `pickle` module for model persistence and storage, as well as discussing Python-specific concerns at deployment time.

## To get the most out of this book

You should have basic understanding of the mathematical principles taught in American primary and high schools. The most complex math required is the understanding of the contents of a matrix and the relation implied by the sigma (sum) symbol. You should have some rudimentary knowledge of Python, including lists, dictionaries, and functions. If you feel deficient in any of these prerequisites, a quick internet search to brush up on the concepts prior to reading should get you ready quickly.

This book is meant as a beginner's text, so the most important prerequisite is an open mind and the drive to learn.

## Download the example code files

You can download the example code files for this book from your account at [www.packt.com](http://www.packt.com). If you purchased this book elsewhere, you can visit [www.packt.com/support](http://www.packt.com/support) and register to have the files emailed directly to you.

You can download the code files by following these steps:

1. Log in or register at [www.packt.com](http://www.packt.com).
2. Select the **SUPPORT** tab.
3. Click on **Code Downloads & Errata**.
4. Enter the name of the book in the **Search** box and follow the onscreen instructions.

Once the file is downloaded, please make sure that you unzip or extract the folder using the latest version of:

- WinRAR/7-Zip for Windows
- Zipeg/iZip/UnRarX for Mac
- 7-Zip/PeaZip for Linux

The code bundle for the book is also hosted on GitHub at <https://github.com/PacktPublishing/Python-Data-Mining-Quick-Start-Guide>. In case there's an update to the code, it will be updated on the existing GitHub repository.

We also have other code bundles from our rich catalog of books and videos available at <https://github.com/PacktPublishing/>. Check them out!

## Download the color images

We also provide a PDF file that has color images of the screenshots/diagrams used in this book. You can download it here: [https://www.packtpub.com/sites/default/files/downloads/9781789800265\\_ColorImages.pdf](https://www.packtpub.com/sites/default/files/downloads/9781789800265_ColorImages.pdf).

## Conventions used

There are a number of text conventions used throughout this book:

A block of code is set as follows, with # used for comment lines:

```
from sklearn.cluster import Method
clus = Method(args*)
# fit to input data
clus.fit(X_input)
# get cluster assignments of X_input
X_assigned = clus.labels_
```

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

```
(base) $ spyder
```

**Bold:** Indicates a new term, an important word, or words that you see onscreen. For example, words in menus or dialog boxes appear in the text like this. Here is an example: "Select **System info** from the **Administration** panel."



Warnings or important notes appear like this.



Tips and tricks appear like this.

## Get in touch

Feedback from our readers is always welcome.

**General feedback:** If you have questions about any aspect of this book, mention the book title in the subject of your message and email us at [customercare@packtpub.com](mailto:customercare@packtpub.com).

**Errata:** Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit [www.packt.com/submit-errata](http://www.packt.com/submit-errata), selecting your book, clicking on the Errata Submission Form link, and entering the details.

**Piracy:** If you come across any illegal copies of our works in any form on the Internet, we would be grateful if you would provide us with the location address or website name. Please contact us at [copyright@packt.com](mailto:copyright@packt.com) with a link to the material.

**If you are interested in becoming an author:** If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit [authors.packtpub.com](http://authors.packtpub.com).

## Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions, we at Packt can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about Packt, please visit [packt.com](http://packt.com).



# 1

## Data Mining and Getting Started with Python Tools

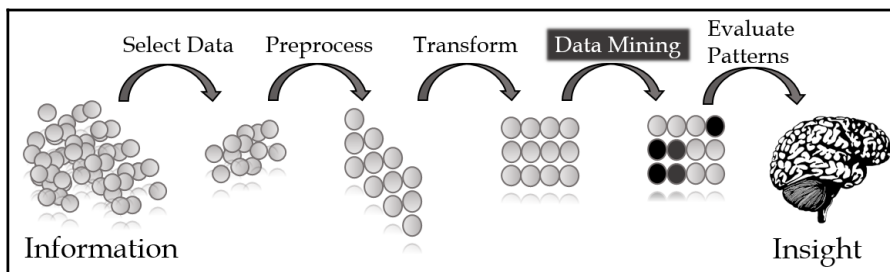
In a sense, data mining is a necessary and predictable response to the dawn of the information age. Indeed, every piece of the modern global economy relies more each year on information and an immense in-stream of data. The path from information pool to actionable insights has many steps. Data mining is typically defined as the pattern and/or trend discovery phase in the pipeline.

This book is a quick-start guide for data mining and will include utilitarian descriptions of the most important and widely used methods, including the mainstays among data professionals such as k-means clustering, random forest prediction, and principal component dimensionality reduction. Along the way, I will give you tips I've learned and introduce helpful scripting tools to make your life easier. Not only will I introduce the tools, but I will clearly describe what makes them so helpful and why you should take the time to learn them.

The first half of the book will cover the nuts and bolts of data collection and preparation. The second half will be more conceptual and will introduce the topics of transformation, clustering, and prediction. The conceptual discussions start in the middle of [Chapter 4, \*Cleaning and Readyng Data for Analysis\*](#), and are written solely as a conversation between myself and the reader. These conversations are ported mostly from the many adhoc training sessions I've done over the years on Intel office marker boards. The last chapter of the book will be on the deployment of these models. This topic is the natural next step for new practitioners and I will provide an introduction and references for when you think you are ready to take the next steps.

The following topics will be covered in this chapter:

- Descriptive, predictive, and prescriptive analytics
- What will and will not be covered in this book
- Setting up Python environments for data mining
- Installing the Anaconda distribution and Conda package manager
- Launching the Spyder IDE
- Launching a Jupyter Notebook
- Installing a high performance Python distribution
- Recommended libraries and how to install



Practitioners should be familiar with the previous data selection, preprocessing, and transformation steps as well as the subsequent pattern and trend evaluation. Knowledge of the full process and an understanding of the goals will orient your data mining efforts in space and keep you aligned with the overall goal.

## Descriptive, predictive, and prescriptive analytics

Practitioners in the field of data analysis usually break down their work into three genres of analytics, given as follows:

- **Descriptive:** Descriptive is the oldest field of analytics study and involves digging deep into the data to hunt down and extract previously unidentified trends, groupings, or other patterns. This was the predominant type of analytics done by the pioneering groups in the field of data mining, and for a number of years the two terms were considered more or less to mean the same thing. However, predictive analytics blossomed in the early 2000s along with the burgeoning field of machine learning, and the many of the techniques that came out of the data mining community proved useful for prediction.
- **Predictive:** Predictive analytics, as the name suggests, focuses on predicting future outcomes and relies on the assumption that past descriptions necessarily lead to future behavior. This concept demonstrates the strong and unavoidable connection between descriptive and predictive analytics. In recent years, industry has naturally taken the next logical step of using prediction to feed into prescriptive solutions.
- **Prescriptive:** Prescriptive analytics relies heavily on customer goals, seeks personalized scoring systems for predictions, and is still a relatively immature field of study and practice. This is accomplished by modeling various response strategies and scoring against the personalized score system.

Please see the following table for a summary:

Type of analytics	Problem statement addressed
Descriptive	What happened?
Predictive	What will happen next?
Prescriptive	How should we respond?

## What will and will not be covered in this book

A quick and dirty description of data mining I hear in the field can be paraphrased as: "Descriptive and predictive analytics with a focus on previously hidden relationships or trends". As such, this book will cover these topics and skip the predictive analytics that focus on automation of obvious prediction, along with the entire field of prescriptive analytics entirely. This text is meant to be a quick start guide, so even the relevant fields of study will only be skimmed over and summarized. Please see the *Recommended reading for further explanation* section for inquiring minds that want to delve deeper into some of the subjects covered in this book.

Preprocessing and data transformation are typically considered to be outside of the data mining category. One of the goals of this book is to provide full working data mining examples, and basic preprocessing is required to do this right. So, this book will cover those topics, before delving in to the more traditional mining strategies.



Throughout this book, I will throw in tips I've learned along my career journey around how to apply data mining to solve real-world problems. I will denote them in a special tip box like this one.

## Recommended readings for further explanation

These books are good for more in-depth discussions and as an introduction to important and relevant topics. I recommend that you start with these if you want to become an expert:

- Data mining in practice:

*Data Mining: Practical Machine Learning Tools and Techniques*, 4th Edition by Ian H. Witten (author), Eibe Frank (author), Mark A. Hall (author), Christopher J. Pal

- Data mining advanced discussion and mathematical foundation:

*Data Mining and Analysis: Fundamental Concepts and Algorithms*, 1st Edition by Mohammed J. Zaki (author), Wagner Meira Jr (author)

- Computer science taught with Python:

*Python Programming: An Introduction to Computer Science*, 3rd Edition by John Zelle (author)

- Python machine learning and analytics:

*Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow*, 2nd Edition Paperback—September 20, 2017 by Sebastian Raschka (author), Vahid Mirjalili (author)

*Advanced Machine Learning with Python Paperback*—July 28, 2016 by John Hearty

## Setting up Python environments for data mining

A computing setup conducive to advanced data mining requires a comfortable development environment and working libraries for data management, analytics, plotting, and deployment. The popular bundled Python distribution from Anaconda is a perfect fit for the job. It is targeted at scientists and engineers, and includes all the required packages to get started. Conda itself is a package manager for maintaining working Python environments and, of course, is included in the bundle. The package manager will allow you to install/remove combinations of libraries into segregated Python environments, all the while reconciling any version dependencies between the distinct libraries.

It includes an integrated development environment called **The Scientific Python Development Environment (Spyder)** and a ready-to-use implementation of Jupyter Notebook interface. Both of these development environments use the interactive Python console called IPython. IPython gives you a live console for scripting. You can run a single line of code, check results, then run another line of code in same console in an interactive fashion. A few trial-and-error sessions with IPython will demonstrate very clearly why these Python tools are so beloved by practitioners working in a rapid prototyping environment.

## Installing the Anaconda distribution and Conda package manager

These tools from Anaconda are available on both Windows and Linux systems. See the following install instructions.

### Installing on Linux

To install the distribution, follow these steps given as follows:

1. First, download the latest installer build from <https://www.anaconda.com/download/#linux>.
2. Then, in the Linux Terminal, pass this bash command:

```
$ bash Anaconda-latest-Linux-x86_64.sh
```

3. Follow the prompts in the terminal and it will begin installing. Once done, you will be asked if you want to allow Conda to be auto-initialized with a `.bashrc` entry. I recommend choosing `N` and activating it manually when needed, just in case you decide to have multiple versions of Conda on your system. In this case, you can launch the Conda prompt by using the following command:

```
$ source /{anaconda3_dir}/bin/activate
```

This will source the Conda activate shell script and call it to activate the `base` environment, which is the default Anaconda Python bundle. Adding new environments will be discussed in the following section on how to install specific libraries. At this point, passing the Python command will open an interactive shell where you can execute Python code line-by-line, as shown in the following code snippet:

```
(base) $ Python
Python 3.7.0 (default, Jun 28 2018, 13:15:42)
[GCC 7.2.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more
information.
>>> import numpy
>>> numpy.random.random(10)
array([0.48489815, 0.80944492, 0.89740441, 0.93031125, 0.71774534,
       0.63817451, 0.93231809, 0.75820457, 0.17550135, 0.62126858])
```

Alternatively, you can execute the code in a stored Python script by using the following command:

```
(base) $ Python script.py
```

## Installing on Windows

To install on Windows, follow the steps given as follows:

1. First, download the executable from <https://conda.io/docs/user-guide/install/windows.html>
2. Then, launch the Anaconda prompt that can be found in a program search from the Windows **Start** menu

Anaconda prompt is a Windows command prompt with all the environment variables set to point to Anaconda. That's it; you are ready to use your base Python environment.

## Installing on macOS

To install on macOS, follow the steps given as follows:

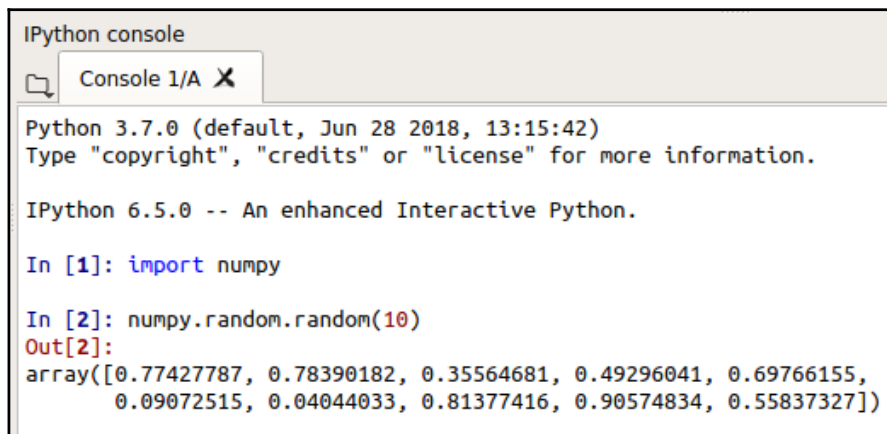
1. First, download the graphical installer from the Anaconda distribution site <https://www.anaconda.com/distribution/>
2. Launch the package and follow the on-screen prompts, which should set up everything you need automatically

## Launching the Spyder IDE

Spyder can be started by passing `spyder` into the Anaconda prompt as follows:

```
(base) $ spyder
```

As mentioned earlier, Spyder uses the IPython interactive console. So, you can pass code line-by-line directly into the console. See the following screenshot for two lines of Python code passed one at a time:



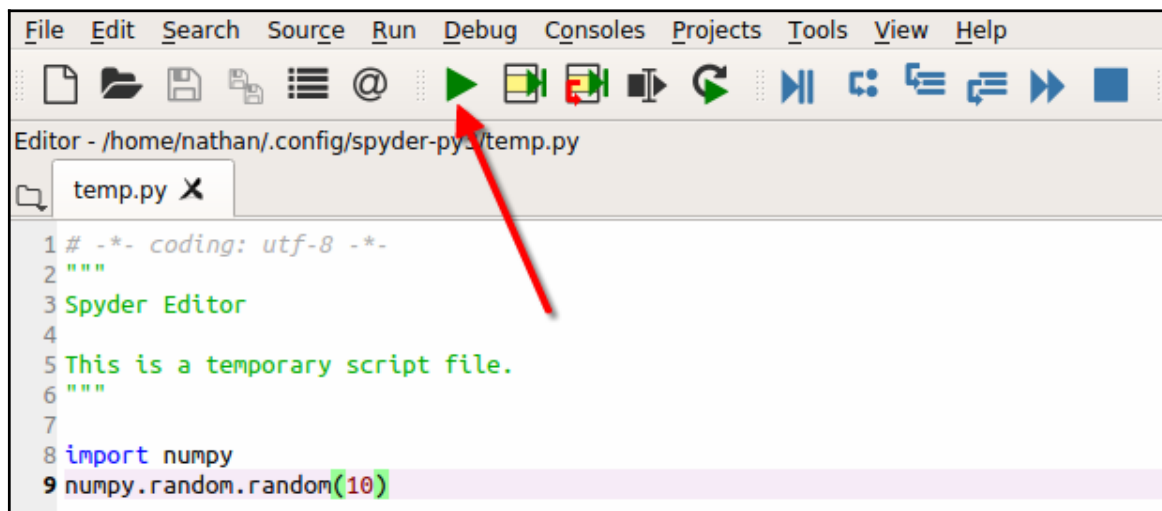
```
Python console
Console 1/A X
Python 3.7.0 (default, Jun 28 2018, 13:15:42)
Type "copyright", "credits" or "license" for more information.

IPython 6.5.0 -- An enhanced Interactive Python.

In [1]: import numpy

In [2]: numpy.random.random(10)
Out[2]:
array([0.77427787, 0.78390182, 0.35564681, 0.49296041, 0.69766155,
       0.09072515, 0.04044033, 0.81377416, 0.90574834, 0.55837327])
```

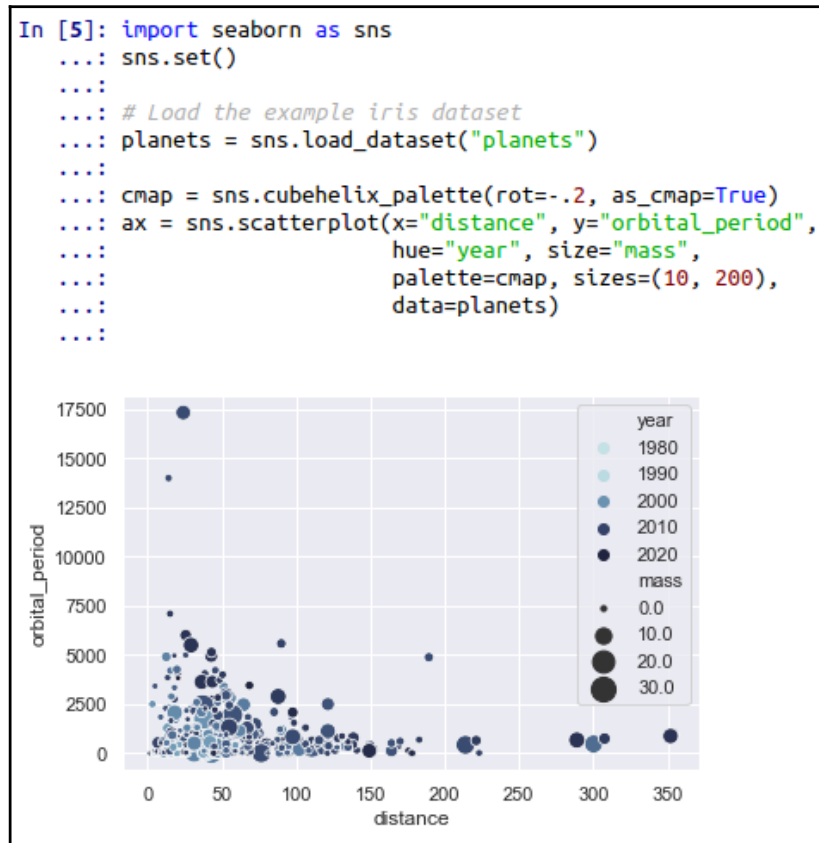
Alternatively, you can edit a script in the editor and execute by pressing the green play button at the top of the IDE. This causes the script to be dumped into the IPython console, then run line-by-line:



```
File Edit Search Source Run Debug Consoles Projects Tools View Help
[Toolbar icons]
Editor - /home/nathan/.config/spyder-py3/temp.py
temp.py X
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4
5 This is a temporary script file.
6 """
7
8 import numpy
9 numpy.random.random(10)
```



The interactive IPython console also can display images and plots inline in the same console window. This is yet another feature that is very convenient for interactive data mining and rapid prototyping of analytics models:

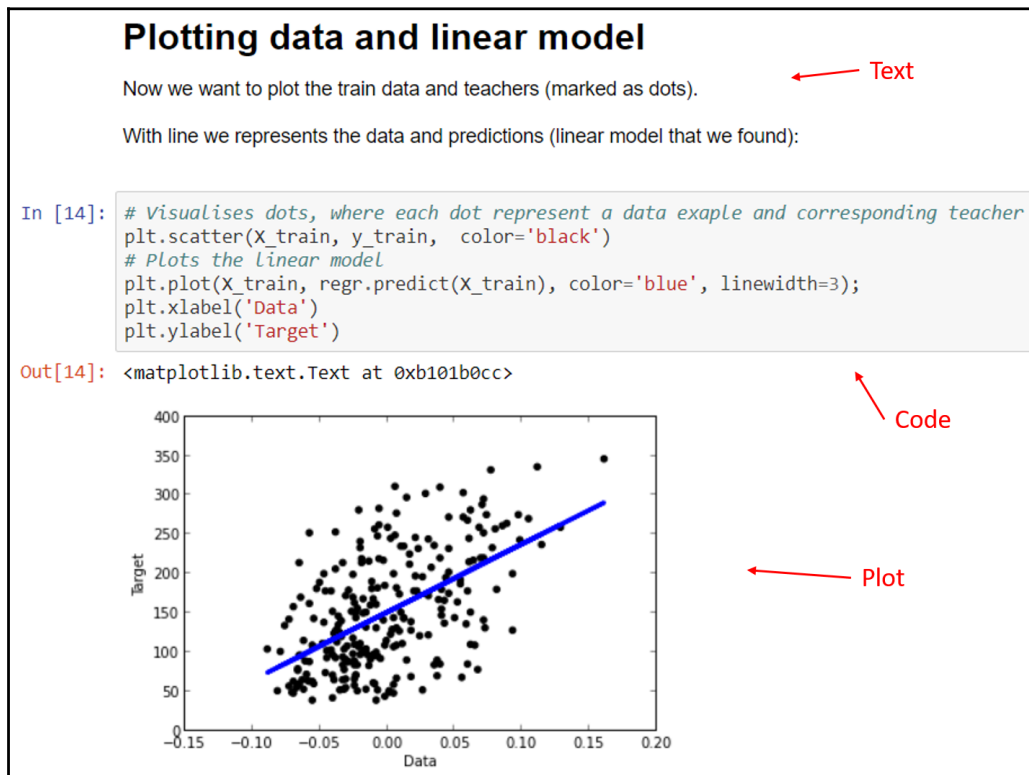


## Launching a Jupyter Notebook

The Jupyter project spun out of the popular IPython Notebook work of the early 2000s. These notebooks provide a visual interface with sequential text and code cells. This allows you to add some text to describe a solution, then follow it with code examples. The Jupyter Notebook also use the IPython console (similar to Spyder), so you have an interactive code interpreter that can plot images inline. Launching the notebook from the Anaconda prompt is simple:

```
(base) $ jupyter notebook
```

The Jupyter project maintains a few basic notebooks. Let's look at a screenshot from one of them, as follows (it can be found at [http://nbviewer.jupyter.org/github/temporaer/tutorial\\_ml\\_gkbionics](http://nbviewer.jupyter.org/github/temporaer/tutorial_ml_gkbionics)):



The concept is self-explanatory if we look at a few examples. The following are recommendations for some relevant and helpful Jupyter Notebooks on data mining and analytics from around the web:

<https://github.com/rasbt/python-machine-learning-book/blob/master/code/ch01/ch01.ipynb>

<http://nbviewer.jupyter.org/github/amplab/datascience-sp14/blob/master/hw2/HW2.ipynb>

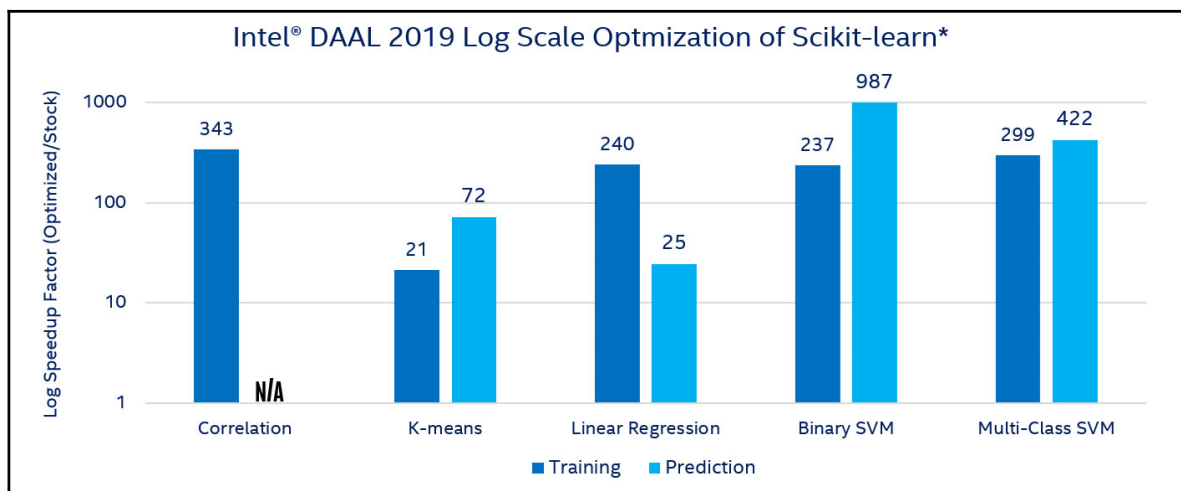
<https://github.com/TomAugspurger/PyDataSeattle/blob/master/notebooks/1.%20Basics.ipynb>

## Installing high-performance Python distribution

Intel Corp has built a bundle of Python libraries with accelerations for **High-Performance Computing (HPC)** on CPUs. The vast majority of the accelerations come with no code changes, because they are snuck in under the hood. All the concepts and libraries introduced in the rest of the book will run faster in the HPC Intel Python environment. Luckily, Intel has a Conda version of their distribution, so you can add it as a new Conda environment via the following few command lines in the Anaconda prompt:

```
(base) $ Conda create -n idp -c channel intelpython3_full Python=3
(base) $ Conda activate idp
```

Full disclosure: I work for Intel, so I won't focus too much on this HPC distribution. I will merely let the performance numbers speak for themselves. See the following graph for raw speedup numbers (optimized versus stock) when using unchanged Scikit-learn code on CPU:



## Recommended libraries and how to install

It's easy to add or remove libraries from the Anaconda prompt. Once you have an the preferred environment activated, the simple `Conda install` command will search the Anaconda cloud repo for a matching package, and will begin download if it exists. Conda will warn if there are version dependencies with your other libraries. Always pay attention to these warnings, so that you know if any other library versions are affected. If, at any time, you need a reminder of what is in your environment, use the `Conda list` command to check package names and versions.

Let's look at some example commands:

1. Create a new environment called `my_env` with Python version 3 using the following command:

```
(base) $ Conda create -n my_env Python=3
```

2. Check whether `my_env` was created successfully by using the following command:

```
(base) $ Conda info --envs
```

You will see the following screen on the execution of the preceding command:

```
(base) nathan@nathan-ThinkPad-Tw1st:~$ conda info --envs
# conda environments:
#
base                * /home/nathan/anaconda3
idp                 /home/nathan/anaconda3/envs/idp
my_env              /home/nathan/anaconda3/envs/my_env
```

3. Activate a new environment by using the following command:

```
(base) $ Conda activate my_env
```

4. Install the `numpy` math library by using the following command:

```
(my_env) $ Conda install numpy
```

5. Use `Conda list` as follows, to check whether a new library was installed or not and all other libraries and versions in `my_env`:

```
(my_env) $ Conda list
```

You will see the following screen on the execution of the preceding command:

```
(my_env) nathan@nathan-ThinkPad-Twist:~$ conda list
# packages in environment at /home/nathan/anaconda3/envs/my_env:
#
# Name                    Version                Build      Channel
blas                      1.0                    mkl
ca-certificates           2018.03.07             0
certifi                   2018.10.15             py37_0
intel-openmp              2019.0                 118
libedit                   3.1.20170329          hb674fdf_2
libffi                    3.2.1                  hd88cf55_4
libgcc-ng                 8.2.0                  hdf63c60_1
libgfortran-ng           7.3.0                  hdf63c60_0
libstdcxx-ng             8.2.0                  hdf63c60_1
mkl                       2019.0                 118
mkl_fft                   1.0.6                  py37h7dd41cf_0
mkl_random                1.0.1                  py37h4414c95_1
ncurses                   6.1                    hf484d3e_0
numpy                     1.15.4                 py37h1d66e8a_0
numpy-base                1.15.4                 py37h81de0dd_0
openssl                   1.1.1                  h7b6447c_0
pip                       18.1                   py37_0
python                    3.7.1                  h0371630_3
readline                  7.0                    h7b6447c_5
setuptools                 40.5.0                 py37_0
sqlite                    3.25.2                 h7b6447c_0
tk                         8.6.8                  hbc83047_0
wheel                     0.32.2                 py37_0
xz                        5.2.4                  h14c3975_4
zlib                      1.2.11                 ha838bed_2
```

## Recommended libraries

If you choose to manage a smaller environment than the full bundle from Anaconda, I recommend the following essential libraries for data mining. They will be used throughout this book:

- `numpy`: The fundamental math library for Python. Brings with it the `numpy` array data structure.
- `scipy`: Provides science and engineering routines built on the base of the `numpy` array. This library also has some good statistical functions.
- `pandas`: Offers relational data tables for storing, labeling, viewing, and manipulating data. You will never look at an array of numbers in the same way for the rest of your career after you've gotten comfortable with `pandas` and its popular data structure, called a `dataframe`.

- `matplotlib`: Python's core visualization library with line and scatter plots, bar and pie charts, histograms and spectrograms, and so on.
- `seaborn`: As statistical visualization library. Built on top of `matplotlib` and much easier to use. You can build complicated visual representations with, in many cases, a single line of code. This library takes `pandas` dataframes as input.
- `statsmodels`: Library focused on statistics functions and statistical testing. For example, it has a `.summary()` function that returns helpful summary stats and information about a model you've applied.
- `scikit-learn`: Python's workhorse machine learning library. It is easy to use and is maintained by an army of developers. The best part is the documentation on <http://scikit-learn.org>. It is so extensive that one could learn the field of machine learning just by reading through the entirety of it.



Editorial: Python has become ubiquitous in the fields of advanced data analysis in the last decade. This is partially due to the scripting nature of the language and approachability to non-programmers, but that is not the whole story. The `pandas`, `scikit-learn`, and `seaborn` libraries are essential to Python's growth in this domain. The power, ease-of-use, well-defined targeted scope, and open source nature of these three libraries are unmatched among free or paid packages. I recommend you learn them inside and out as you embark on a career in data mining.

## Summary

This chapter introduced the contents of the book and covered getting started with your software environment. It also covered how to get high-speed Python and popular libraries such as `pandas`, `scikit-learn`, and `seaborn`. After reading this chapter and setting your environment, you should be ready to follow along with the demonstrations throughout the rest of the book.

# 2

## Basic Terminology and Our End-to-End Example

The philosophy behind a quick-start guide is that the topic at hand is best learned by doing. In this chapter, I will present a quick overview of important vocabulary, concepts, and terminology that you need to get started, and then jump directly into a full end-to-end working example of data mining in Python. Later chapters will flesh out the steps in the working example in more detail.

The following topics will be covered in this chapter:

- Basic data terminology
- Basic statistics
- An end-to-end example of data mining in Python

### **Basic data terminology**

This section is meant to be a quick overview of the terms that you should know before you get started. This list is very streamlined and is not exhaustive. Please refer to the suggested reading in [Chapter 1, \*Data Mining and Getting Started with Python Tools\*](#), for wider coverage of domain-specific terminology.