



# Python Data Analytics

With Pandas, NumPy, and Matplotlib

—  
*Second Edition*  
—

Fabio Nelli

Apress®

# **Python Data Analytics**

**With Pandas, NumPy,  
and Matplotlib**

**Second Edition**

**Fabio Nelli**

**Apress®**

## ***Python Data Analytics***

Fabio Nelli  
Rome, Italy

ISBN-13 (pbk): 978-1-4842-3912-4

<https://doi.org/10.1007/978-1-4842-3913-1>

ISBN-13 (electronic): 978-1-4842-3913-1

Library of Congress Control Number: 2018957991

Copyright © 2018 by Fabio Nelli

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr

Acquisitions Editor: Todd Green

Development Editor: James Markham

Coordinating Editor: Jill Balzano

Cover image designed by Freepik ([www.freepik.com](http://www.freepik.com))

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit [www.springeronline.com](http://www.springeronline.com). Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail [rights@apress.com](mailto:rights@apress.com), or visit <http://www.apress.com/rights-permissions>.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at [www.apress.com/9781484239124](http://www.apress.com/9781484239124). For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

# Table of Contents

About the Author .....xvii

About the Technical Reviewer .....xix

Chapter 1: An Introduction to Data Analysis ..... 1

    Data Analysis ..... 1

    Knowledge Domains of the Data Analyst ..... 3

        Computer Science ..... 3

        Mathematics and Statistics ..... 4

        Machine Learning and Artificial Intelligence ..... 5

        Professional Fields of Application..... 5

    Understanding the Nature of the Data ..... 5

        When the Data Become Information..... 6

        When the Information Becomes Knowledge ..... 6

        Types of Data..... 6

    The Data Analysis Process ..... 6

        Problem Definition ..... 8

        Data Extraction ..... 9

        Data Preparation..... 10

        Data Exploration/Visualization..... 10

        Predictive Modeling..... 12

        Model Validation ..... 13

        Deployment ..... 13

    Quantitative and Qualitative Data Analysis ..... 14

    Open Data ..... 15

    Python and Data Analysis..... 17

    Conclusions..... 17

## TABLE OF CONTENTS

<b>Chapter 2: Introduction to the Python World .....</b>	<b>19</b>
Python—The Programming Language.....	19
Python—The Interpreter.....	21
Python 2 and Python 3 .....	23
Installing Python .....	23
Python Distributions .....	24
Using Python.....	26
Writing Python Code .....	28
IPython.....	35
PyPI—The Python Package Index.....	39
The IDEs for Python .....	40
SciPy .....	46
NumPy .....	47
Pandas.....	47
matplotlib .....	48
Conclusions.....	48
<b>Chapter 3: The NumPy Library .....</b>	<b>49</b>
NumPy: A Little History.....	49
The NumPy Installation .....	50
Nddarray: The Heart of the Library.....	50
Create an Array .....	52
Types of Data .....	53
The dtype Option .....	54
Intrinsic Creation of an Array .....	55
Basic Operations .....	57
Arithmetic Operators .....	57
The Matrix Product .....	59
Increment and Decrement Operators .....	60
Universal Functions (ufunc) .....	61
Aggregate Functions .....	62

Indexing, Slicing, and Iterating .....	62
Indexing .....	63
Slicing .....	65
Iterating an Array .....	67
Conditions and Boolean Arrays .....	69
Shape Manipulation .....	70
Array Manipulation .....	71
Joining Arrays .....	71
Splitting Arrays .....	72
General Concepts .....	74
Copies or Views of Objects .....	75
Vectorization .....	76
Broadcasting .....	76
Structured Arrays .....	79
Reading and Writing Array Data on Files .....	82
Loading and Saving Data in Binary Files .....	82
Reading Files with Tabular Data .....	83
Conclusions .....	84
<b>Chapter 4: The pandas Library—An Introduction .....</b>	<b>87</b>
pandas: The Python Data Analysis Library .....	87
Installation of pandas .....	88
Installation from Anaconda .....	88
Installation from PyPI .....	89
Installation on Linux .....	90
Installation from Source .....	90
A Module Repository for Windows .....	90
Testing Your pandas Installation .....	91
Getting Started with pandas .....	92
Introduction to pandas Data Structures .....	92
The Series .....	93

TABLE OF CONTENTS

- The DataFrame ..... 102
  - The Index Objects ..... 112
- Other Functionalities on Indexes..... 114
  - Reindexing ..... 114
  - Dropping ..... 117
  - Arithmetic and Data Alignment..... 118
- Operations Between Data Structures..... 120
  - Flexible Arithmetic Methods ..... 120
  - Operations Between DataFrame and Series..... 121
- Function Application and Mapping..... 122
  - Functions by Element ..... 123
  - Functions by Row or Column..... 123
  - Statistics Functions ..... 125
- Sorting and Ranking ..... 126
- Correlation and Covariance..... 129
- “Not a Number” Data ..... 131
  - Assigning a NaN Value..... 131
  - Filtering Out NaN Values..... 132
  - Filling in NaN Occurrences ..... 133
- Hierarchical Indexing and Leveling ..... 134
  - Reordering and Sorting Levels ..... 137
  - Summary Statistic by Level..... 138
- Conclusions..... 139
- Chapter 5: pandas: Reading and Writing Data..... 141**
  - I/O API Tools ..... 141
  - CSV and Textual Files ..... 142
  - Reading Data in CSV or Text Files ..... 143
    - Using RegExp to Parse TXT Files ..... 146
    - Reading TXT Files Into Parts..... 148
    - Writing Data in CSV ..... 150

# About the Technical Reviewer



**Raul Samayoa** is a senior software developer and machine learning specialist with many years of experience in the financial industry. An MSc graduate from the Georgia Institute of Technology, he's never met a neural network or dataset he did not like. He's fond of evangelizing the use of DevOps tools for data science and software development.

Raul enjoys the energy of his hometown of Toronto, Canada, where he runs marathons, volunteers as a technology instructor with the University of Toronto coders, and likes to work with data in Python and R.



## CHAPTER 1

# An Introduction to Data Analysis

In this chapter, you begin to take the first steps in the world of data analysis, learning in detail about all the concepts and processes that make up this discipline. The concepts discussed in this chapter are helpful background for the following chapters, where these concepts and procedures will be applied in the form of Python code, through the use of several libraries that will be discussed in just as many chapters.

## Data Analysis

In a world increasingly centralized around information technology, huge amounts of data are produced and stored each day. Often these data come from automatic detection systems, sensors, and scientific instrumentation, or you produce them daily and unconsciously every time you make a withdrawal from the bank or make a purchase, when you record various blogs, or even when you post on social networks.

But what are the data? The data actually are not information, at least in terms of their form. In the formless stream of bytes, at first glance it is difficult to understand their essence if not strictly the number, word, or time that they report. Information is actually the result of processing, which, taking into account a certain dataset, extracts some conclusions that can be used in various ways. This process of extracting information from raw data is called *data analysis*.

The purpose of data analysis is to extract information that is not easily deducible but that, when understood, leads to the possibility of carrying out studies on the mechanisms of the systems that have produced them, thus allowing you to forecast possible responses of these systems and their evolution in time.

Starting from a simple methodical approach on data protection, data analysis has become a real discipline, leading to the development of real methodologies generating *models*. The model is in fact the translation into a mathematical form of a system placed under study. Once there is a mathematical or logical form that can describe system responses under different levels of precision, you can then make predictions about its development or response to certain inputs. Thus the aim of data analysis is not the model, but the quality of its *predictive power*.

The predictive power of a model depends not only on the quality of the modeling techniques but also on the ability to choose a good dataset upon which to build the entire data analysis process. So the *search for data*, their *extraction*, and their subsequent *preparation*, while representing preliminary activities of an analysis, also belong to data analysis itself, because of their importance in the success of the results.

So far we have spoken of data, their handling, and their processing through calculation procedures. In parallel to all stages of processing of data analysis, various methods of *data visualization* have been developed. In fact, to understand the data, both individually and in terms of the role they play in the entire dataset, there is no better system than to develop the techniques of graphic representation capable of transforming information, sometimes implicitly hidden, in figures, which help you more easily understand their meaning. Over the years lots of display modes have been developed for different modes of data display: the *charts*.

At the end of the data analysis process, you will have a model and a set of graphical displays and then you will be able to predict the responses of the system under study; after that, you will move to the test phase. The model will be tested using another set of data for which you know the system response. These data are, however, not used to define the predictive model. Depending on the ability of the model to replicate real observed responses, you will have an error calculation and knowledge of the validity of the model and its operating limits.

These results can be compared with any other models to understand if the newly created one is more efficient than the existing ones. Once you have assessed that, you can move to the last phase of data analysis—*deployment*. This consists of implementing the results produced by the analysis, namely, implementing the decisions to be taken based on the predictions generated by the model and the associated risks.

Data analysis is well suited to many professional activities. So, knowledge of it and how it can be put into practice is relevant. It allows you to test hypotheses and to understand more deeply the systems analyzed.

# Knowledge Domains of the Data Analyst

Data analysis is basically a discipline suitable to the study of problems that may occur in several fields of applications. Moreover, data analysis includes many tools and methodologies that require good knowledge of computing, mathematical, and statistical concepts.

A good data analyst must be able to move and act in many different disciplinary areas. Many of these disciplines are the basis of the methods of data analysis, and proficiency in them is almost necessary. Knowledge of other disciplines is necessary depending on the area of application and study of the particular data analysis project you are about to undertake, and, more generally, sufficient experience in these areas can help you better understand the issues and the type of data needed.

Often, regarding major problems of data analysis, it is necessary to have an interdisciplinary team of experts who can contribute in the best possible way in their respective fields of competence. Regarding smaller problems, a good analyst must be able to recognize problems that arise during data analysis, inquire to determine which disciplines and skills are necessary to solve these problems, study these disciplines, and maybe even ask the most knowledgeable people in the sector. In short, the analyst must be able to know how to search not only for data, but also for information on how to treat that data.

## Computer Science

Knowledge of computer science is a basic requirement for any data analyst. In fact, only when you have good knowledge of and experience in computer science can you efficiently manage the necessary tools for data analysis. In fact, every step concerning data analysis involves using calculation software (such as IDL, MATLAB, etc.) and programming languages (such as C ++, Java, and Python).

The large amount of data available today, thanks to information technology, requires specific skills in order to be managed as efficiently as possible. Indeed, data research and extraction require knowledge of these various formats. The data are structured and stored in files or database tables with particular formats. XML, JSON, or simply XLS or CSV files, are now the common formats for storing and collecting data, and many applications allow you to read and manage the data stored on them. When it comes to extracting data contained in a database, things are not so immediate, but you need to know the SQL query language or use software specially developed for the extraction of data from a given database.

Moreover, for some specific types of data research, the data are not available in an explicit format, but are present in text files (documents and log files) or web pages, and shown as charts, measures, number of visitors, or HTML tables. This requires specific technical expertise for the parsing and the eventual extraction of these data (called *web scraping*).

So, knowledge of information technology is necessary to know how to use the various tools made available by contemporary computer science, such as applications and programming languages. These tools, in turn, are needed to perform data analysis and data visualization.

The purpose of this book is to provide all the necessary knowledge, as far as possible, regarding the development of methodologies for data analysis. The book uses the Python programming language and specialized libraries that provide a decisive contribution to the performance of all the steps constituting data analysis, from data research to data mining, to publishing the results of the predictive model.

## Mathematics and Statistics

As you will see throughout the book, data analysis requires a lot of complex math during the treatment and processing of data. You need to be competent in all of this, at least to understand what you are doing. Some familiarity with the main statistical concepts is also necessary because all the methods that are applied in the analysis and interpretation of data are based on these concepts. Just as you can say that computer science gives you the tools for data analysis, so you can say that the statistics provide the concepts that form the basis of data analysis.

This discipline provides many tools to the analyst, and a good knowledge of how to best use them requires years of experience. Among the most commonly used statistical techniques in data analysis are

- Bayesian methods
- Regression
- Clustering

Having to deal with these cases, you'll discover how mathematics and statistics are closely related. Thanks to the special Python libraries covered in this book, you will be able to manage and handle them.

## Machine Learning and Artificial Intelligence

One of the most advanced tools that falls in the data analysis camp is machine learning. In fact, despite the data visualization and techniques such as clustering and regression, which should help you find information about the dataset, during this phase of research, you may often prefer to use special procedures that are highly specialized in searching patterns within the dataset.

Machine learning is a discipline that uses a whole series of procedures and algorithms that analyze the data in order to recognize patterns, clusters, or trends and then extracts useful information for data analysis in an automated way.

This discipline is increasingly becoming a fundamental tool of data analysis, and thus knowledge of it, at least in general, is of fundamental importance to the data analyst.

## Professional Fields of Application

Another very important point is the domain of competence of the data (its source—biology, physics, finance, materials testing, statistics on population, etc.). In fact, although analysts have had specialized preparation in the field of statistics, they must also be able to document the source of the data, with the aim of perceiving and better understanding the mechanisms that generated the data. In fact, the data are not simple strings or numbers; they are the expression, or rather the measure, of any parameter observed. Thus, better understanding where the data came from can improve their interpretation. Often, however, this is too costly for data analysts, even ones with the best intentions, and so it is good practice to find consultants or key figures to whom you can pose the right questions.

## Understanding the Nature of the Data

The object of study of data analysis is basically the data. The data then will be the key player in all processes of data analysis. The data constitute the raw material to be processed, and thanks to their processing and analysis, it is possible to extract a variety of information in order to increase the level of knowledge of the system under study, that is, one from which the data came.

## When the Data Become Information

Data are the events recorded in the world. Anything that can be measured or categorized can be converted into data. Once collected, these data can be studied and analyzed, both to understand the nature of the events and very often also to make predictions or at least to make informed decisions.

## When the Information Becomes Knowledge

You can speak of knowledge when the information is converted into a set of rules that helps you better understand certain mechanisms and therefore make predictions on the evolution of some events.

## Types of Data

Data can be divided into two distinct categories:

- Categorical (nominal and ordinal)
- Numerical (discrete and continuous)

*Categorical data* are values or observations that can be divided into groups or categories. There are two types of categorical values: *nominal* and *ordinal*. A nominal variable has no intrinsic order that is identified in its category. An ordinal variable instead has a predetermined order.

*Numerical data* are values or observations that come from measurements. There are two types of numerical values: *discrete* and *continuous* numbers. Discrete values can be counted and are distinct and separated from each other. Continuous values, on the other hand, are values produced by measurements or observations that assume any value within a defined range.

## The Data Analysis Process

Data analysis can be described as a process consisting of several steps in which the raw data are transformed and processed in order to produce data visualizations and make predictions thanks to a mathematical model based on the collected data. Then, data

**F**

Financial data, 552  
 Flexible arithmetic methods, 120–121  
 Fonts, LaTeX, 539

**G**

Gradient theory, 523  
 Graphics Processing Unit (GPU), 353  
 Grouping, 11  
 Group iteration  
   chain of transformations, 222, 224  
   functions on groups  
     mark() function, 224–225  
     quantiles() function, 224  
 GroupBy object, 222

**H**

Handwriting recognition  
   digits dataset, 475–478  
   handwritten digits, matplotlib  
     library, 478  
   learning and predicting, 478, 480, 482  
   OCR software, 473  
   scikit-learn, 474–475  
   svc estimator, 480  
   TensorFlow, 480  
   validation set, six digits, 479  
 Health data, 550  
 Hierarchical indexing  
   arrays, 136–137  
   DataFrame, 135  
   reordering and sorting levels, 137–138  
   stack() function, 136  
   statistic levels, 138  
   structure, 134  
   two-dimensional structure, 134

**I**

IDEs, *see* Interactive development environments (IDEs)  
 Image analysis  
   concept of, 521  
   convolutions, 523  
   definition, 507  
   edge detection, 522, 525  
     blackandwhite.jpg  
       image, 526–529, 531  
     black and white system, 525  
     filters function, 528  
     gradients.jpg image, 532  
     gray gradients, 525  
     Laplacian and Sobel filters, 531  
     results, 528  
     source code, 530  
   face detection, 532  
   gradient theory, 523  
   OpenCV (*see* Open Source Computer Vision (OpenCV))  
   operations, 508  
   representation of, 522  
 Indexing functionalities  
   arithmetic and data  
     alignment, 118, 120  
   dropping, 117–118  
   reindexing, 114, 116  
 Integration, 47  
 Interactive development environments (IDEs)  
   Eclipse (pyDev), 41–42  
   Komodo, 45  
   Liclipse, 43–46  
   NinjaIDE, 44–45  
   Spyder, 41  
   Sublime, 42–43

## INDEX

Interactive programming language, [20](#)  
Interfaced programming language, [20](#)  
Internet of Things (IoT), [353](#)  
Interpreted programming language, [20](#)  
Interpreter  
    characterization, [21](#)  
    Cython, [22](#)  
    Jython, [22](#)  
    PVM, [21](#)  
    PyPy, [22](#)  
    tokenization, [21](#)

IPython  
    and IPython QtConsole, [233–234](#)  
    Jupyter project logo, [37](#)  
    Notebook, [39](#), [474](#)  
        DataFrames, [420](#)  
    QtConsole, [38](#)  
    shell, [36](#)  
    tools of, [35](#)

Iris flower dataset  
    Anderson Iris Dataset, [316](#)  
    IPython QtConsole, [316](#)  
    Iris setosa features, [318–319](#)  
    length and width,  
        petal, [319–320](#)  
    matplotlib library, [318](#)  
    PCA decomposition, [320](#)  
    target attribute, [317](#)  
    types of analysis, [316](#)  
    variables, [319–320](#)

## J

JavaScript D3 Library  
    bar chart, [454](#)  
    CSS definitions, [450–451](#)  
    data-driven documents, [449](#)  
    HTML importing library, [450](#)

IPython Notebooks, [449](#)  
Jinja2 library, [451–453](#)  
    pandas dataframe, [453](#)  
    render() function, [453](#)  
    require.config() method, [450](#)  
    web chart creation, [450](#)

Jinja2 library, [451–453](#)  
Jython, [22](#)

## K

K-nearest neighbors classification  
    decision boundaries, [325–326](#)  
    2D scatterplot, sepals, [324](#)  
    predict() function, [323](#)  
    random.permutation(), [323](#)  
    training and testing set, [322](#)

## L

LaTeX  
    accents, [540–547](#)  
    fonts, [539](#)  
    fractions, binomials, and stacked  
        numbers, [538–539](#)  
    with IPython Notebook  
        in Markdown Cell, [537](#)  
        in Python 2 Cell, [538](#)  
    with matplotlib, [537](#)  
    radicals, [539](#)  
    subscripts and superscripts, [538](#)  
    symbols  
        arrow symbols, [540](#), [545–546](#)  
        big symbols, [542](#)  
        binary operation and relation  
            symbols, [542–543](#)  
        Delimiters, [540–541](#)  
        Hebrew, [541](#)



- lowercase Greek, 540
  - miscellaneous symbols, 540
  - standard function names, 542
  - uppercase Greek, 541
  - Learning phase, 378
  - Liclipse, 43–46
  - Linear regression, 12
  - Line chart
    - annotate(), 274
    - arrowprops kwarg, 274
    - Cartesian axes, 273
    - color codes, 270–271
    - data points, 267
    - different series, 269
    - gca() function, 273
    - Greek characters, 272
    - LaTeX expression, 274
    - line and color styles, 270
    - mathematical expressions, 275
    - mathematical function, 268
    - pandas, 276
    - plot() function, 268
    - set\_position() function, 273
    - xticks() and yticks() functions, 271
  - Linux distribution, 90
  - LOD cloud diagram, 16
  - Logistic regression, 12
- ## M
- Machine learning (ML), 5
    - algorithm development process, 313
    - deep learning, 351
    - diabetes dataset, 327–328
    - features/attributes, 314
    - Iris flower dataset, 316
    - learning problem, 314
    - linear/least square regression
      - coef\_ attribute, 329
      - fit() function, 329
      - linear correlation, 330
      - parameters, 328
      - physiological factors and
        - progression of diabetes, 332–333
        - single physiological factor, 330
      - schematization of, 352
      - supervised learning, 314
      - SVM (*see* Support vector machines (SVMs))
      - training and testing set, 315
      - unsupervised learning, 314–315
  - Mapping
    - adding values, 201–202
    - inplace option, 204
    - rename() function, 204
    - renaming, axes, 202, 204
    - replacing values, 199, 201
  - Mathematical expressions with LaTeX,
    - see* LaTeX
  - MATLAB, 17
  - matplotlib, 48
  - matplotlib library
    - architecture
      - artist layer, 236–238
      - backend layer, 236
      - functions and tools, 235
      - layers, 235
      - pylab and pyplot, 238–239
      - scripting layer (pyplot), 238
    - artist layer
      - graphical representation, 237
      - hierarchical structure, 236
      - primitive and composite, 237
    - graphical representation, 231, 233
    - LaTeX, 232
    - NumPy, 246

## INDEX

Matrix product, [60](#)

Merging operation

    DataFrame, [183–184](#)

    dataframe objects, [183](#)

    index, [187](#)

    join() function, [187–188](#)

    JOIN operation, [182](#)

    left\_index/right\_index

        options, [187](#)

    left join, right join and

        outer join, [186](#)

    left\_on and right\_on, [185, 187](#)

    merge() function, [183, 184](#)

Meteorological data, [409](#)

    Adriatic Sea and Po Valley, [410](#)

        cities, [412](#)

        Comacchio, [413](#)

        image of, [411](#)

        mountainous areas, [410](#)

        reference standards, [412](#)

        TheTimeNow website, [413](#)

    climate, [409](#)

    data source

        JSON file, [414](#)

        Weather Map site, [414](#)

    IPython Notebook

        chart representation, [425, 429, 431](#)

        CSV files, [421](#)

        DataFrames, [422, 432](#)

        humidity function, [433–435](#)

        linear regression, [431](#)

        matplotlib library, [423](#)

        Milan, [423](#)

        read\_csv() function, [421](#)

        result, [423](#)

        shape() function, [422](#)

        SVR method, [428–429](#)

        temperature, [424, 426–427, 432](#)

    Jupyter Notebook, [415](#)

        access internal data, [417](#)

        command line, [415](#)

        dataframe, [419–420](#)

        extraction procedures, [418](#)

        Ferrara, [416](#)

        JSON file, [416](#)

        json.load() function, [415](#)

        parameters, [419](#)

        prepare() function, [420](#)

    RoseWind (*see* RoseWind)

    wind speed, [441](#)

Microsoft excel files

    dataframe, [162](#)

    data.xls, [160, 162](#)

    internal module xlrd, [160](#)

    read\_excel() function, [161](#)

MongoDB, [178–179](#)

Multi Layer Perceptron (MLP)

    artificial networks, [360](#)

    evaluation of, [404](#)

    experimental data, [404](#)

    hidden layers, [397](#)

    IPython session, [387](#)

    learning phase, [389](#)

    model definition, [387](#)

    test phase and accuracy

        calculation, [395, 402](#)

Musical data, [553](#)

## N

Natural Language Toolkit (NLTK)

    bigrams and collocations, [498](#)

    common\_contexts() function, [493](#)

    concordance() function, [493](#)

    corpora, [488](#)

    downloader tool, [489](#)

- fileids() function, 491
  - HTML pages, text, 501
  - len() function, 491
  - library, 489
  - macbeth variable, 491
  - Python library, 488
  - request() function, 502
  - selecting words, 497
  - sentimental analysis, 502
  - sents() function, 492
  - similar() function, 494
  - text, network, 500
  - word frequency, 494
    - macbeth variable, 495
    - most\_common() function, 494
    - nltk.download() function, 495
    - nltk.FreqDist() function, 494
    - stopwords, 495
    - string() function, 496
  - word search, 493
  - Ndarray, 47
    - array() function, 51–53
    - data, types, 53–54
    - dtype (data-type), 50, 54
    - intrinsic creation, 55–57
    - type() function, 51–52
  - NOSE MODULE, 91
  - “Not a Number” data
    - filling, NaN occurrences, 133
    - filtering out NaN
      - values, 132–133
    - NaN value, 131–132
  - NumPy library
    - array manipulation (*see* Array manipulation)
    - basic operations
      - aggregate functions, 62
      - arithmetic operators, 57–59
      - increment and decrement
        - operators, 60–61
      - matrix product, 59–60
      - ufunc, 61
    - broadcasting
      - compatibility, 77
      - complex cases, 78–79
      - operator/function, 76
    - BSD, 50
    - conditions and Boolean arrays, 69
    - copies/views of objects, 75
    - data analysis, 49
    - indexing, 63
      - bidimensional array, 64
      - monodimensional ndarray, 63
      - negative index value, 63
    - installation, 50
    - iterating an array, 67–69
    - ndarray (*see* Ndarray)
    - Numarray, 49
    - python language, 49
    - reading and writing array data, 82
    - shape manipulation, 70–71
    - slicing, 65–66
    - structured arrays, 79
    - vectorization, 76
- ## O
- Object-oriented programming language, 20
  - OCR, *see* Optical Character Recognition
    - (OCR) software
  - Open data, 15–16
  - Open data sources, 353
    - climatic data, 552
    - demographics
      - IPython Notebook, 446
      - matplotlib, 449